Heap Overflows

- What is on the heap? _____
- Also of interest: the Data section: _____ and _____ variables

```
// Heap Buffer Overflow code 1
#define BUFSIZE 16
#define OVERSIZE 8 /* overflow buf2 by OVERSIZE bytes */
   int main()
   {
      u_long diff;
      char *buf1 = (char *)malloc(BUFSIZE);
      char *buf2 = (char *)malloc(BUFSIZE);

      diff = (u_long)buf2 - (u_long)buf1;
      printf("buf1 = %p, buf2 = %p, diff = 0x%x bytes\n",
             buf1, buf2, diff);

      memset(buf2, 'A', BUFSIZE-1);
      buf2[BUFSIZE-1] = '\0';

      printf("before overflow: buf2 = %s\n", buf2);

      memset(buf1, 'B', (u_int)(diff + OVERSIZE));
      printf("after overflow: buf2 = %s\n", buf2);

      return 0;
   }
```

```
// File Pointer Overwrite
#define BUFSIZE 16
int main(int argc, char **argv)
{
  FILE *tmpfd;
  static char buf[BUFSIZE], *tmpfile;

  tmpfile = ”/tmp/vulprog.tmp";
  printf("before: tmpfile = %s\n", tmpfile);

  printf("Enter one line of data to put in %s: ",
         tmpfile);
  gets(buf);

  printf("\nafter: tmpfile = %s\n", tmpfile);

  tmpfd = fopen(tmpfile, "w");
  if (tmpfd == NULL) exit(ERROR);

  fputs(buf, tmpfd);
  fclose(tmpfd);
}
```

- What does the code print?
- Address manipulation: a _____ can modify a _____

Exploiting a Heap Overflow

- Targets: _____, _____
- Function Pointers
  - Used for _____, _____
  - Implementation: stored in _____.

    Upon call, copied into the _____!
- Return-to-libc: Directly call _____! Assumes _____

Vulnerabilities

Obviously vulnerable functions:

Replacements (bounded):

Defenses

- Programmer: _____
- Compiler: _____
- System: _____

Programmer-level Solutions

- Use a _____ programming language: _____
- Libraries that _____: LibSafe
- Program better! Problem: _____

Compiler-level solutions

- Always _____
- StackGuard
  - _____: random value stored in _____
  - Check _____
- StackShield: _____
- _____ (ASLR)
  - Recall: (even w/ _____) need to guess the _____
  - Starting stack/heap/etc. address set randomly in a range _____

System Solutions

- Grow the stack backwards? _____
  - Doesn't prevent _____
- Non-_____ memory
  - AMD: _____ bit; Intel: ____ bit. Mark each page of memory
  - Prevents _____ inside that page