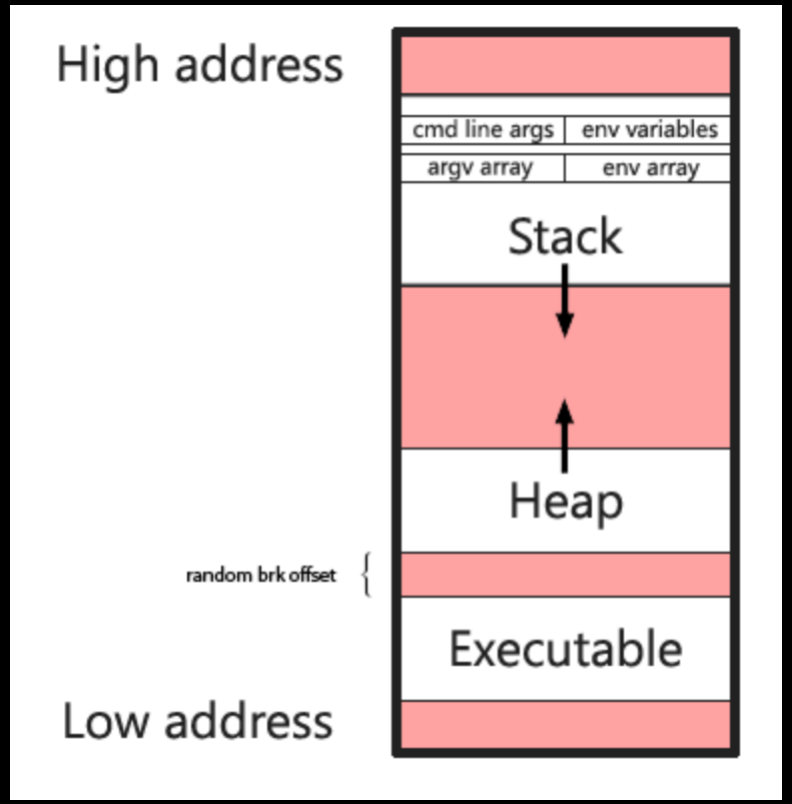
Announcements

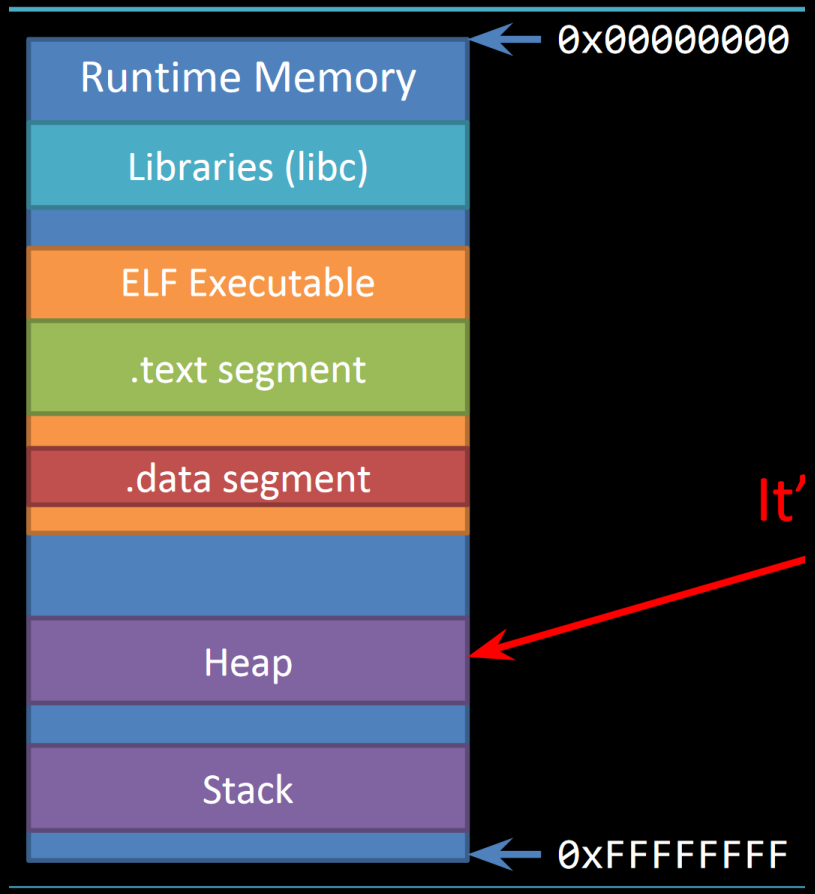
- Website: cs595g.lockshaw.io
- Challenge Expectations
- Remaining Course Overview

Heap Exploitation

The Heap

malloc, free

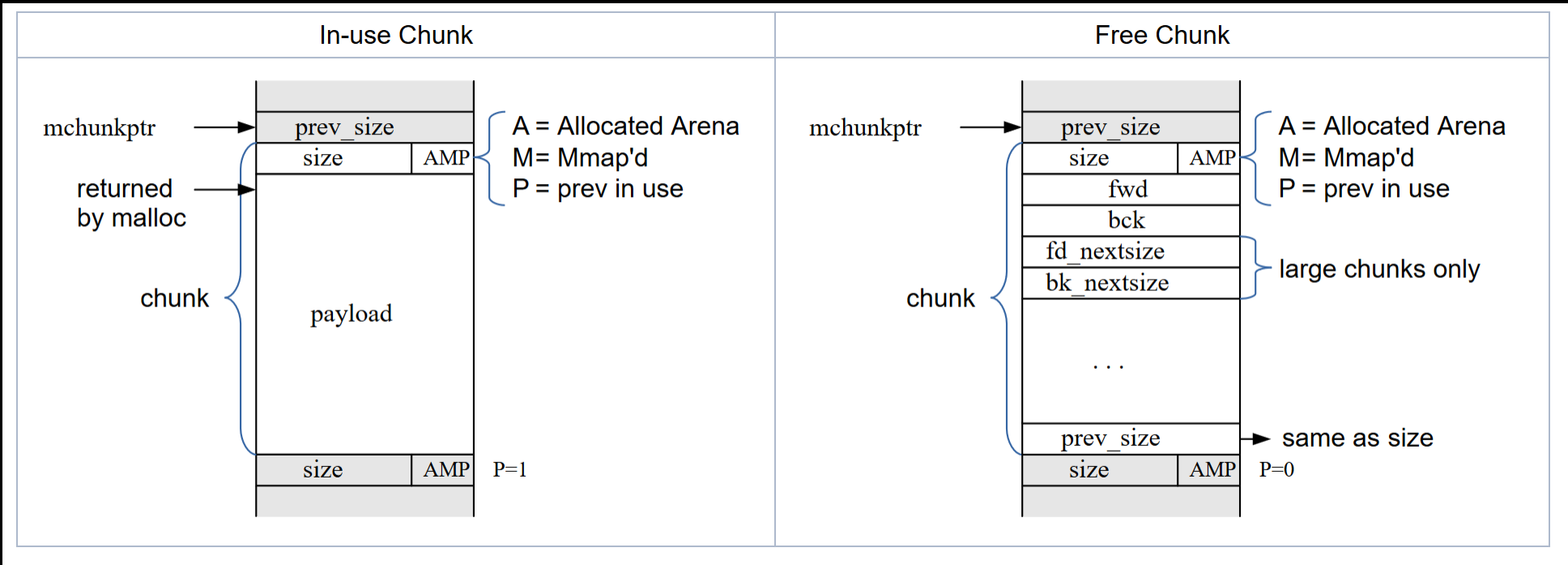




Heap Goals

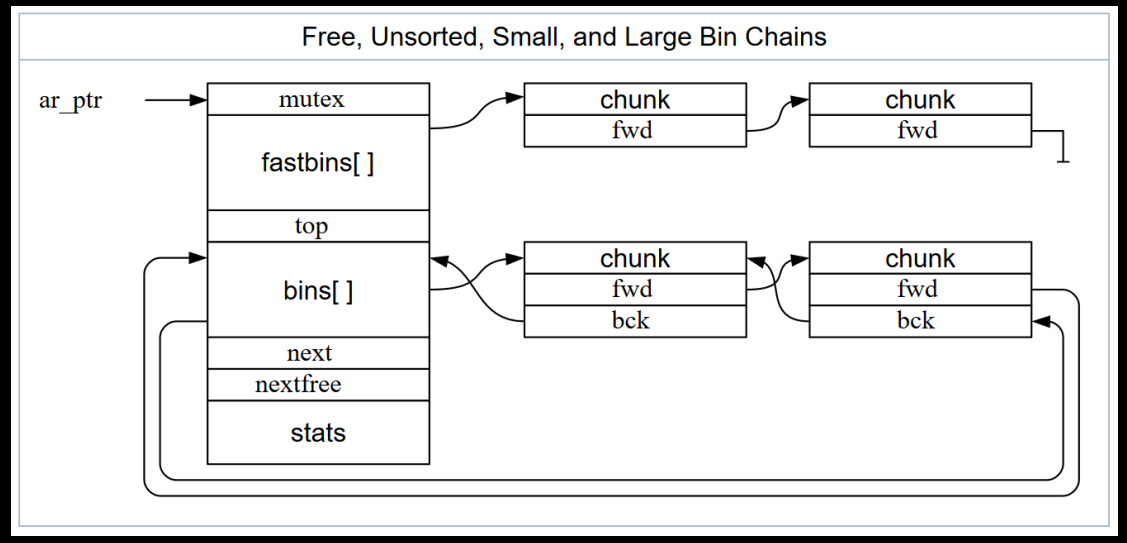
- Fast
- Fragmentation
- Thread Safety
- Space-Efficient

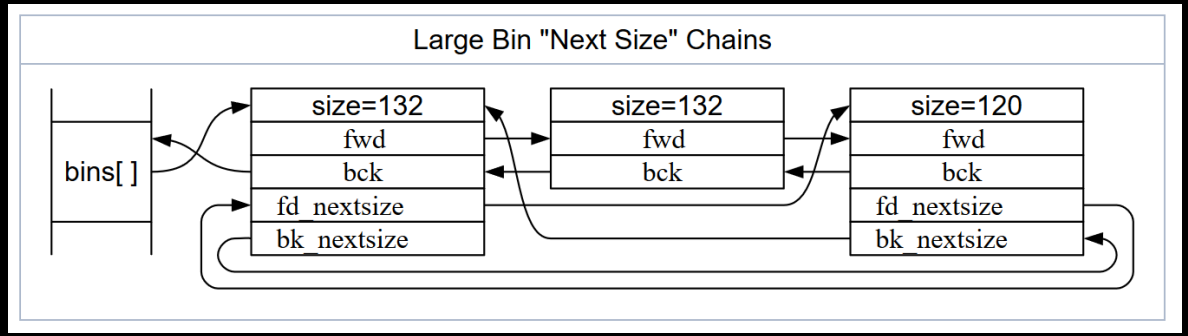
Chunk Layout

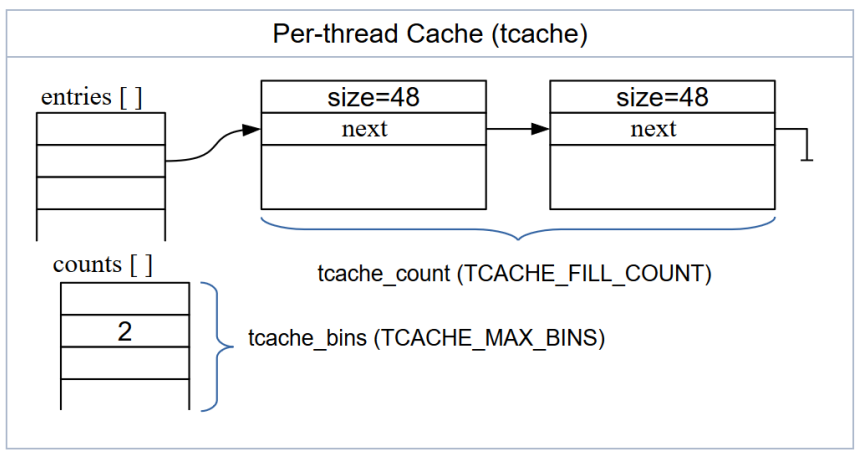


Bins

- Fastbins
- Smallbins
- Largebins
- Unsorted Bin
- Tcache







Breaking the Heap

- Goal:
  - Arbitrary write
- General Strategy:
  - Corrupt Next (fd), Prev (bk) pointers
  - Corrupt Size header

Use After Free (UAF)

```c
#include <stdlib.h>
#include <stdio.h>

int main() {
  void *a = malloc(0x50);
  fgets(a, 0x50, stdin);
  free(a);
  fgets(a, 0x50, stdin);
  void *b = malloc(0x50);
  void *c = malloc(0x50);
  fgets(c, 0x50, stdin);

  return;
}
```

Heap Overflow

```c
#include <stdlib.h>
#include <stdio.h>

int main() {
  void *a = malloc(0x50);
  void *b = malloc(0x50);
  void *c = malloc(0x50);
  free(b);
  free(c);
  gets(a);
  void *d = malloc(0x50);
  void *e = malloc(0x50);
  fgets(c, 0x50, stdin);

  return;
}
```

Forging Chunks

```c
#include <stdlib.h>
#include <stdio.h>

int main() {
  void *a = malloc(0x1000);
  printf("%p", a);
  fgets(a, 0x1000, stdin);
  void *x;
  scanf("%p", &x);
  free(x);

  void *b = malloc(0x50);
  void *c = malloc(0x50);
  fgets(c, 0x50, stdin);

  return;
}

```

Now What?

- __malloc_hook
- __free_hook
- GOT
- FILE *
- Any other function pointers

Tools

- Pwngdb (link on website)