Heap 0

- Use the current working directory to bypass the GetCode()
- GetCode() will use the first 4 bytes from the flag file
- Create flag.txt file on server to control the value of the code in a different directory
- We need to try and call line #35
- Memcopy function allows us to overflow the buffer into the chunks that contain a struct
- Top chunk represents heap memory that has not been allocated. Malloc chops memory from this chunk

- The bottom most chunk we can ignore
- f -> fp get addr of puts (Line #29)
  f -> is_valid = set_code 0s stores 4 bytes of flag
- d becaus the command str we need to use in order to call the shell
  Line # 35
- Must overwrite fp to be system
- Create a symlink? for the flag

Challenge uses ASLR !!

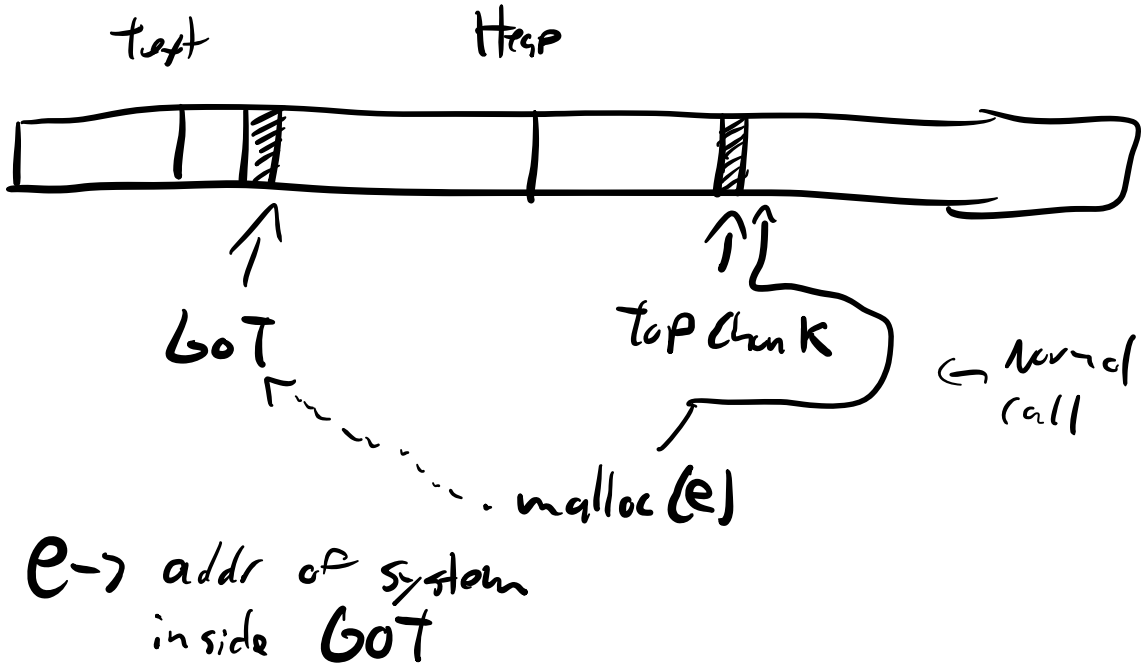
ASLR must be aligned for pages

- libc_system's first 3 nibbles never change
- Dont need to change the higher bytes

- Need to overwrite the first 3 bytes
- Compare the outputs to see if your guess is correct
- Could try a previously found lower 3 bytes of a system address

Heap Meta Data

- Can manipulate heap meta data to make malloc start at the Global Offset Table



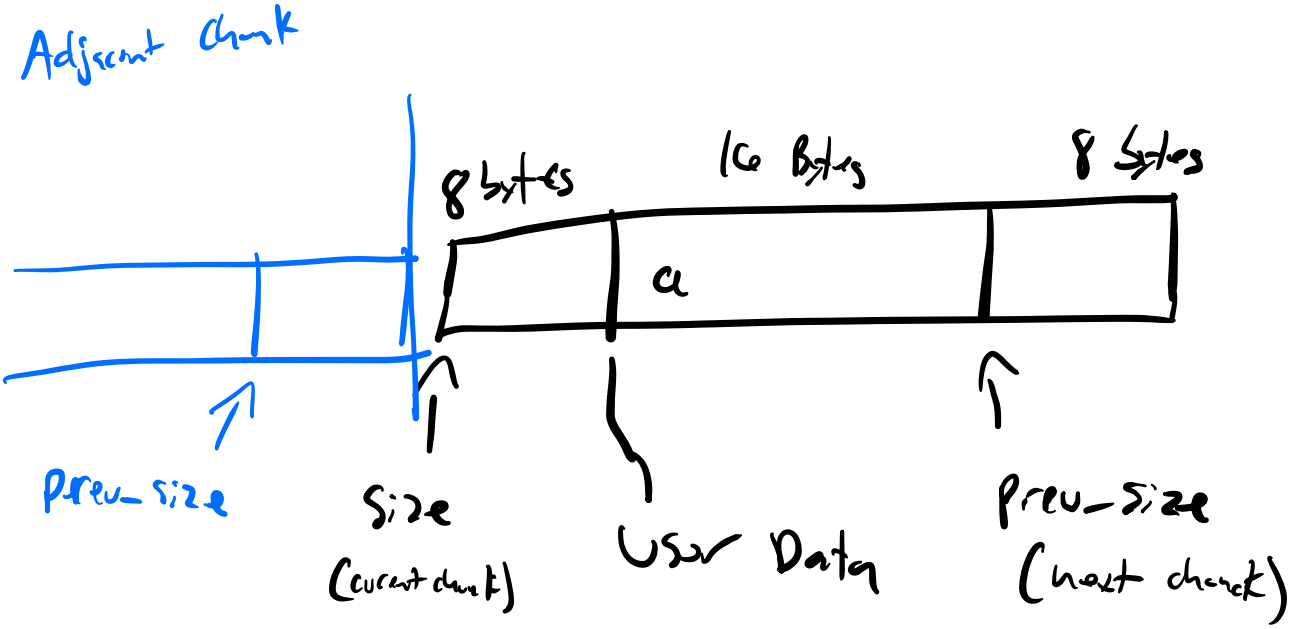
- If we can trick malloc to allocate where we want it can be very powerful

Heap chunks must be aligned to 16 bytes

Meta Data = 16 bytes

User Data = 16 bytes

Chunk = 32 bytes



Heap Bins

When you free a chunk of heap memory, it actually gets put in a bin for possible reuse

If a chunk is binned, it can be reused if another allocation is made. It

will take memory from the top of the binned chunk

Bin Categories

- Small
- Large
- Fast
- Tcache
- Unsorted

Tcache: Set of Bins specific to each thread in the program

Before tcache, locks were used to prevent collisions of allocations. However it causes a hit to performance

★ Can hold at most 7 Allocations ★

Tcache Bin is essentially a Single Linked list that tracks chunks freed.

Tcache is specific for each thread

Tcache bins hold the same size of freed memory. If different, its set to a different tcache bin

Chunks in tcache are continously marked as "In-Use" to avoid the consolidation of chunks.

Tcache chunks ordered by when they were free'd

The most recently freed chunk in tcache bin will be reallocated

★ Last-In-First-Out List ★