Heap Overflow

Binary Exploitation

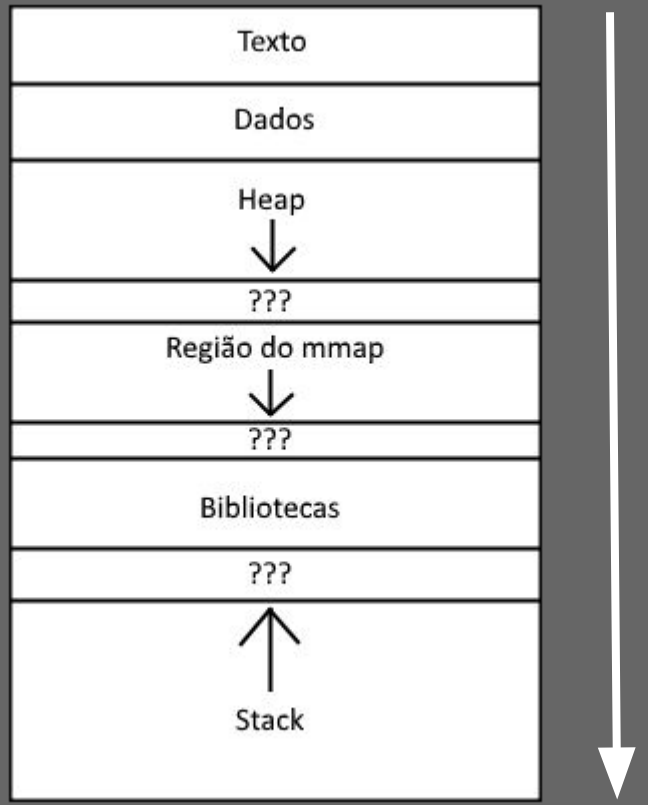
O que é?

Heap ~~Overflow~~

- Não é igual a estrutura de dados
- Depende da implementação
  - dlmalloc (Doug Lea malloc)
  - ptmalloc (pthreads malloc)
  - glibc malloc (GNU C library's malloc)
- Rapidez vs fragmentação

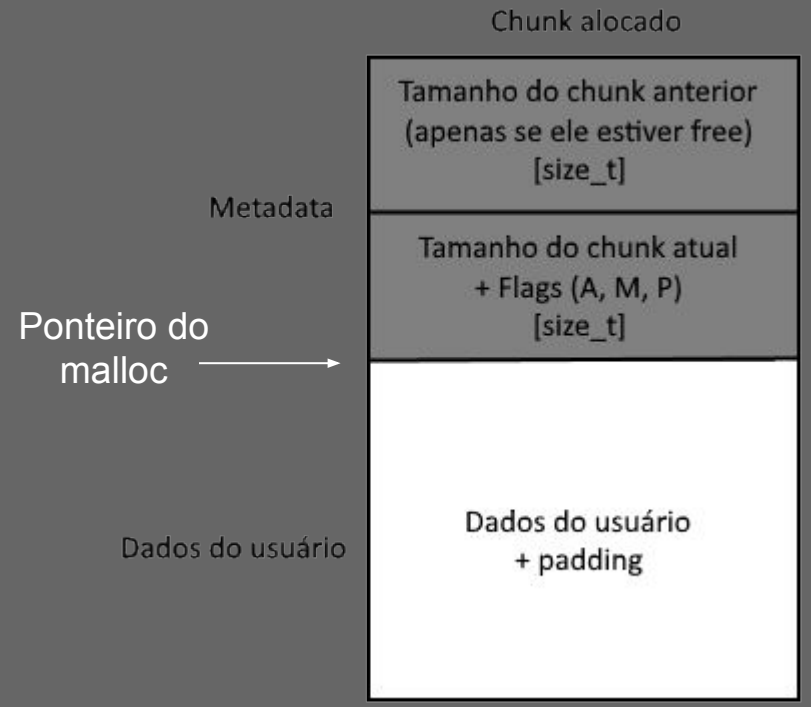
Implementação Padrão da Heap no glibc

- Características:
  - Chunks
  - Arenas
  - Bins



Chunks

- Metadata + dados do usuário
  - 32 bits: alinhamento em 8 bytes
  - 64 bits: alinhamento em 16 bytes

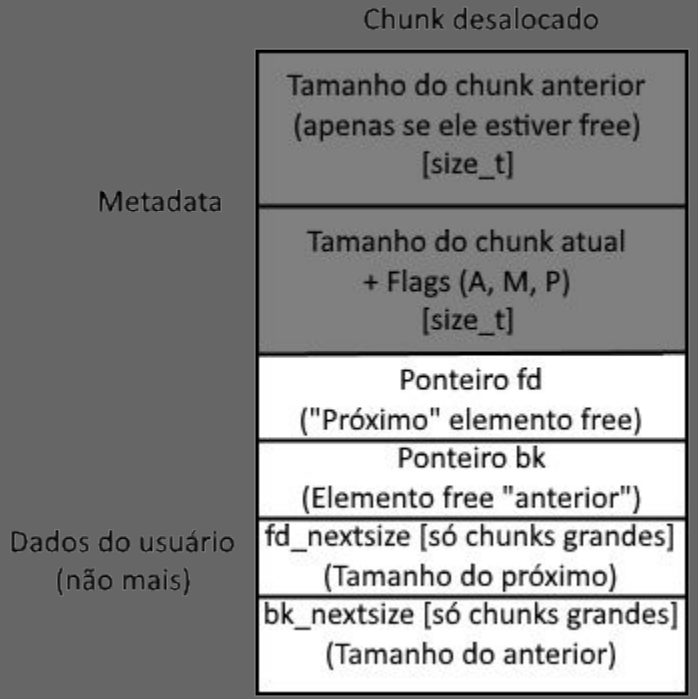


Regras simplificadas de alocação

- Chunk free no meio grande o suficiente
- Espaço livre no topo da heap
- Pede pro kernel aumentar o tamanho da heap
- Devolve NULL pro malloc

Free’d chunks

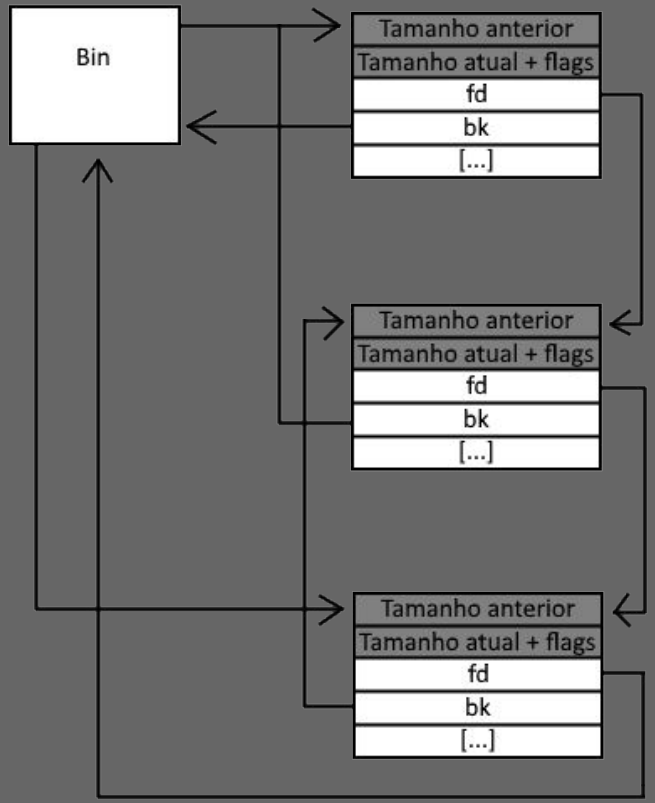
- O que acontece quando eu dou free?



Realocação de memória e bins

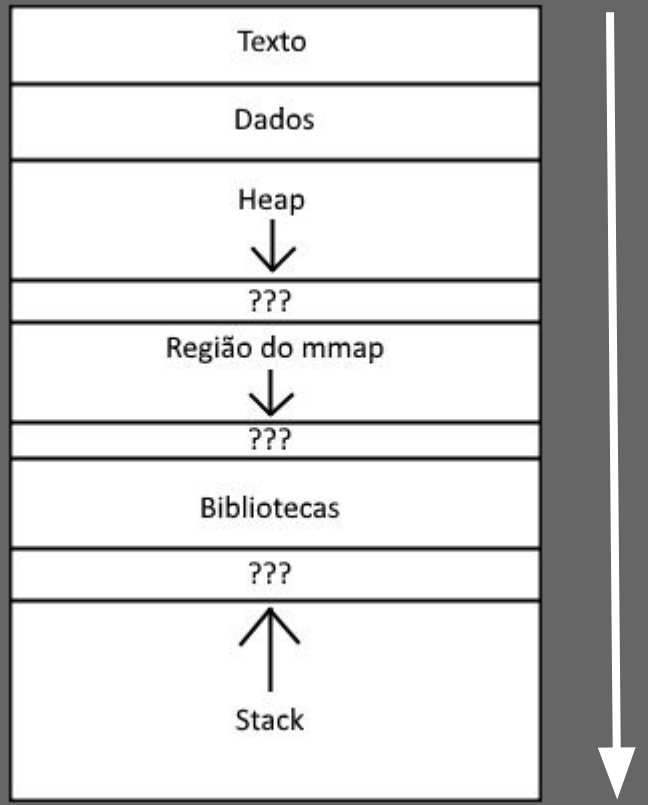
- O que são bins?
  - Fast bins
  - Unsorted bin
  - Small bins
  - Large bins
  - Tcache bins

- Como é a realocação?



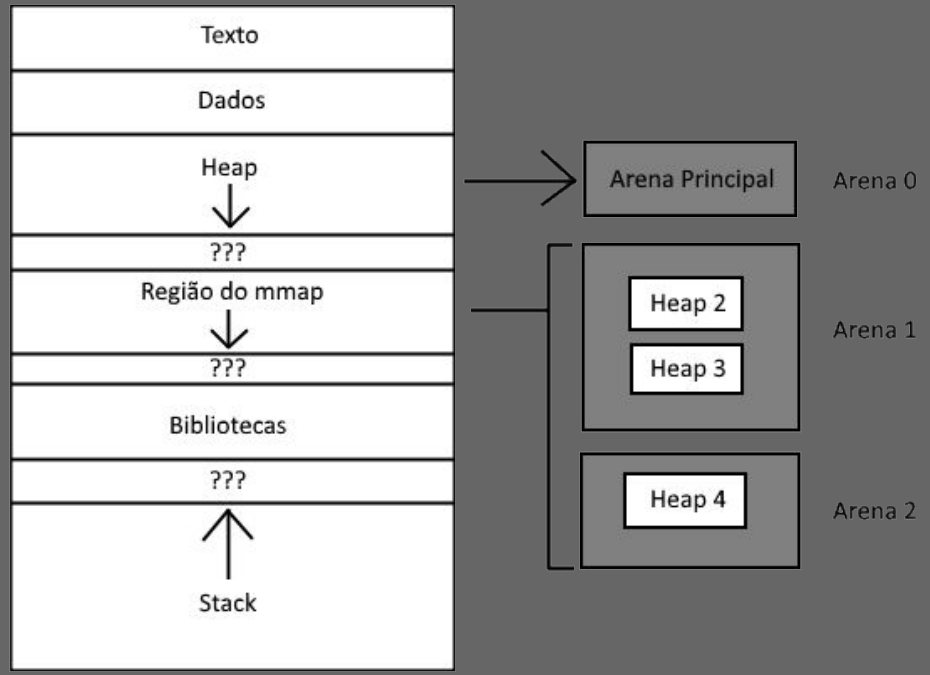
Alocação de memória no fim da heap

- Se houver espaço
- Se não houver espaço
  - sbrk e brk
  - mmap
- Alocações gigantes
  - 32-bits: entre 128KB e 512KB
  - 64-bits: 32MB
  - Pode aumentar dinamicamente
  - Via mmap



Arenas

- Processos multithread
  - Global mutex
  - Arenas
- Máximo de arenas
  - 32-bits: 2x nº de núcleos da CPU
  - 64-bits: 8x nº de núcleos da CPU
- Arenas e múltiplas heaps



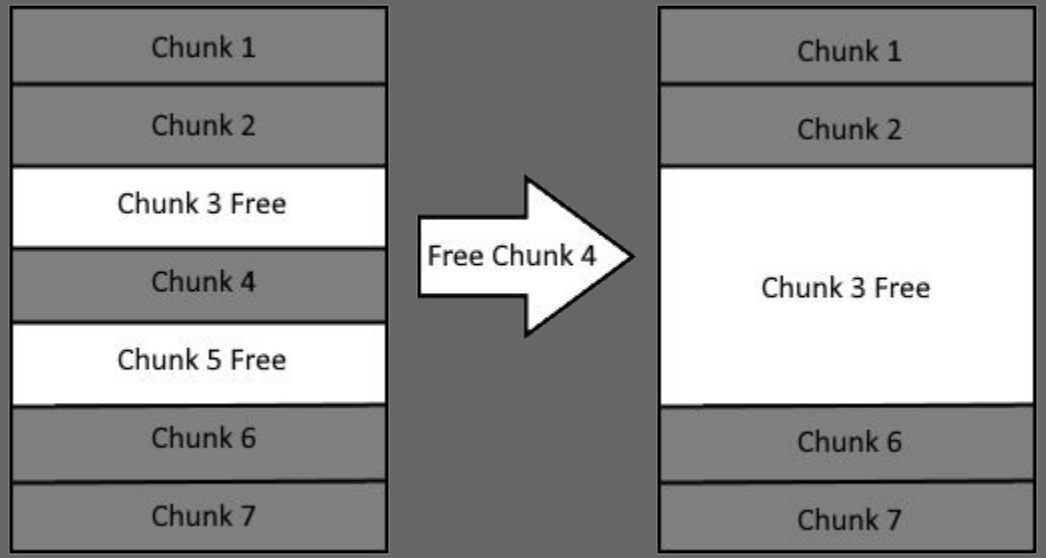
Heaps secundárias

- Heap primária
  - sbrk
- Outras heaps
  - mmap
    - 32-bits: 1MB
    - 64-bits: 64MB
    - PROT_NONE
  - mprotect
    - PROT_NOME -> PROT_READ | PROT_WRITE

Metadata dos chunks

- Flags
  - A (Allocated arena)
  - M (Mmap'd chunk)
  - P (Previous chunk in use)
- Tamanho do chunks + flags?
  - 00010001 (17)
  - 00100100 (36)
  - 01000010 (66)

Junção de chunks

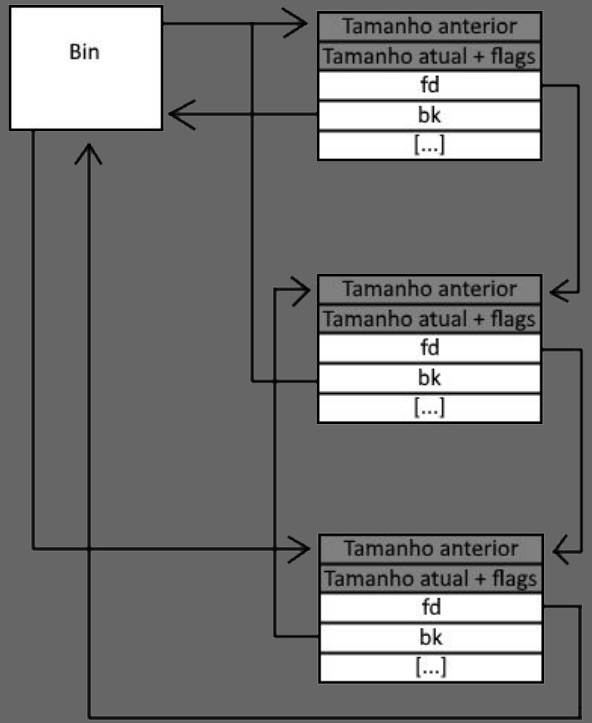


A função free()

- Sanity checks
  - Não faz nada se for NULL
  - Alocação não alinhada em 8/16 bytes
  - Tamanho do chunk estranho
    - Muito grande ou pequeno
    - Tamanho não alinhado em 8/16 bytes
    - Tamanhos que quase dão overflow
  - Alocação fora da arena
  - Próximo chunk não tem a flag P

Os tipos de bins

- Tipos básicos:
  - Small bins
  - Large bins
  - Unsorted bins
- Tipos para otimização:
  - Fast bins
  - Tcache bins (glibc)



Índices dos bins básicos

- Bin[0] = Não usado
- Bin[1] = Unsorted bin
- Bin[2] até Bin[63] = Small bin
- Bin[64] até Bin[126] = Large bin

Small bins

- 62 bins
  - 32-bits: chunk < 512 bytes
  - 64-bits: chunk < 1024 bytes
- 1 bin para cada tamanho
  - 32-bits: 16, 24, …, 504
  - 64-bits: 32, 48, …, 1008
- Acesso rápido
- Chunks pequenos = mais comum

Large bins

- 63 bins
  - O que não couber nas small bins
- Indexação por intervalos
  - 1 a 32 (32): Intervalos de 64B
  - 33 a 48 (16): Intervalos de 512B
  - 49 a 56 (8): Intervalos de 4KB
  - 57 a 60 (4): Intervalos de 32KB
  - 61 a 62 (2): Intervalos de 256KB
  - 63 (1): O que sobrar (>1MB)

Unsorted bin

- Frees frequentemente são sucedidos por mallocs de tamanho igual
- Unsorted precede small e large
- Opera a partir do malloc

Fast Bins

- 10 bins
  - 32-bits: 16, 24, …, 88 bytes
  - 64-bits: 32, 48, …, 176 bytes
- Simplesmente encadeada
- Não dá free de verdade
- Frees “reais” acontecem quando:
  - Mallocs maiores que a small bin
  - Frees acima de 64KB
  - Funções especiais

Tcache (Per-thread cache) Bins

- Otimizar multithreading
- Reduzir o custo do “lock contention”
- Adicionado pelo glibc 2.26
- 64 bins simplesmente encadeadas
- Máximo de 7 chunks de tamanho igual
- Tamanho dos chunks:
  - 32-bits: 12 a 516 bytes
  - 64-bits: 24 a 1032 bytes

Funcionamento das tcache bins

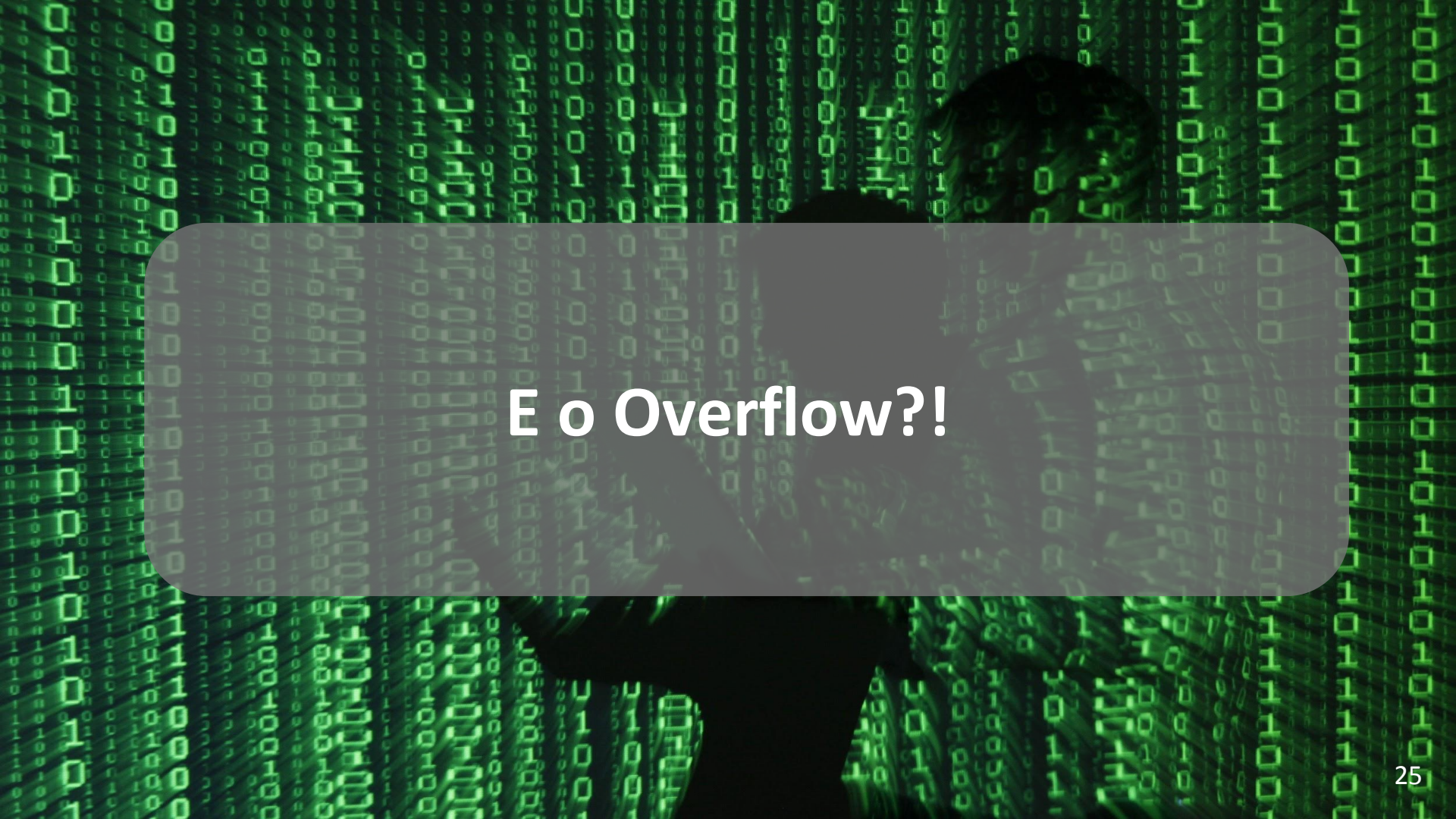
- Inserção
  - Não da free de verdade
  - Se o bin tiver cheio, ignora o tcache
- Alocação
  - Apenas se forem de tamanhos “exatos”
  - Se a bin procurada estiver vazia
    atravessa a “heap lock” e pega
    até 7 chunks daquele tamanho
    e retorna o último

Revisando malloc

- Tcache
- Muito grande = mmap
- Fast bin
- Small bin
- Faz os frees de verdade
- Unsorted bin
- Large bin
- Cria um novo
- Retorna NULL

Revisando free

- Nada se for NULL
- Converte o ponteiro para o começo do chunk
- Sanity checks
- Tcache
- Se tiver a flag M, munmap
- Arena heap lock
- Fast bin
- Frees e junções
- Unsorted bin


E o Overflow?!

Heap Overflow (finalmente)

- Sobrescrição de dados
- Ponteiros
- Auxílio para outros exploits na heap

Outros exploits de heap

- Use-after-free
- Double free
- Exploit do comportamento das bins

GANESH

Grupo de Segurança da Informação
ICMC / USP - São Carlos, SP
http://ganesh.icmc.usp.br/
ganesh@icmc.usp.br