ASSIGNMENT 4

YELLOW-PAGES

- Overwrite →`next` in another entry by overflowing the buffer.
- You don't need to deal with heap *metadata* yet, but heap *layout* matters.
- Then, proceed as in `phonebook`.

GHOSTBUSTERS

- You can `free` arbitrary addresses.
- Build a fake chunk in one of your allocations that you then `free` ($\Longrightarrow$ *House of Spirit*).
- Make the fake chunk large enough, so you get a really big allocation on the stack when you allocate again.
- Write your ROP chain as usual.

TINYTCASH

- `checksec`: No RELRO, no PIE
- Goal: Overwrite GOT entry with win function
- Straightforward tcache exploit

STRINGBINS

- `checksec`: everything enabled (except `_FORTIFY_SOURCE`)
- Trivial stack and heap pointer leaks (via `%p`)
- The `strings_p` entry is not zeroed after `free` → use-after-free
- `change` clears the buffer first, so we can't use it on `strings_p`
- Make `malloc`/`realloc` return the same chunk twice by overwriting →`fd`
- Allocate a string on top of `strings_p`
- Overwrite the pointers to point to the stack (during `alloc`)
- Leak the libc from the stack, then write a ROP chain

DESREVER

- `magic` to do ”magic”.
- This interprets the scores in sorted order as shellcode and runs it.
- We told you sorted shellcode was going to happen 😉

ON LIBC GOT

- Used to select optimized versions of functions based on available hardware at runtime (remember magic8ball's `GLIBC_TUNABLES`?)
- This means entries vary by hardware
- Some of the unoptimized implementations call other functions, when the optimized version does not
- *Some* entry will usually work, but it may not be the same as on your machine — try different offsets
- Typical trick: most print functions (`puts`, `printf`, ...) will call `strlen` or `strnlen`.