CPSC 525/625: Principles of Computer Security (F23)
Instructor: Ryan Henry <ryan.henry@ucalgary.ca>

UNIVERSITY OF CALGARY

Name: Alex Stevenson UCID: 30073617

Worksheet 0x08

In /srv/cheech.c you will find the source code for a silly "ping"-like program. The cheech program takes one argument (an arbitrary-length string) that it forwards (via a pipe) to a companion program, /usr/bin/chong, which merely echos the provided string back to cheech. From here, cheech compares the string it sent to the one it received and indicates the result of the comparison.

I can assure you that this program was written for a very good reason and is very useful, alas over the past few hours of chilling with cheech and /usr/bin/chong, I seem to have just plum forgot what the reason is or what makes the program useful. Oh well, Imma go grab a snack.

Oh, yeah. I almost forgot. Here's a snippet from the source for /usr/bin/chong. Specifically, this is the function that actually reads the payload from cheech and echos it back. Notice how it most definitely allocates the correct amount of memory and most definitely does not overflow the buffer. Which is good, because /usr/bin/chong also handles some sensitive data, and it'd be downright catastrophic if that sensitive data got clobbered or—worse—leaked into the hands of my CPSC525/625 students!

```
// also found in /srv/chong.c
int chong()
{
    size_t payload_len;
    size_t bytes_read = read(STDIN_FILENO, &payload_len, sizeof(payload_len));

    char * payload = malloc(payload_len);
    if (payload == NULL)
    {
        return 0;
    }
    bytes_read = read(STDIN_FILENO, payload, payload_len);

    size_t bytes_written = write(STDOUT_FILENO, payload, payload_len);
    free(payload);
    payload = NULL;

    return 1;
}
```

CPSC 525/625: Principles of Computer Security (F23)
Instructor: Ryan Henry <ryan.henry@ucalgary.ca>

UNIVERSITY OF CALGARY

Name: Alex Stevenson UCID: 30073617

Your task

1. Modify `/srv/cheech.c` to trick `/usr/bin/chong` into leaking the sensitive data it handles. (Hint: This is basically a simplified version of the Heartbleed vulnerability.)

   Write a brief description of what the vulnerability is and how you exploited it, and submit your modified `cheech.c` to the dropbox alongside this worksheet.

The vulnerability has cheech tell chong how many bytes the payload will be before sending a payload of that many bytes. Chong reads the data and then sends cheech back a payload of the specified bytes. By having cheech tell chong that the payload is much larger than the data actually is, chong will attempt to send more data than it was given and end up sending extra arbitrary data from memory.

I exploited this by increasing the payload length by 999, such that
`size_t payload_len = strlen(cheech)+1000;`

After receiving the chong, I printed out the data character by character and got a ton of extra data that was not part of the cheech.

```
for (int i = 0; i < sizeof(chong); i++) {
    printf("%c", chong[i]);
}
```