Homework 10

1. *The `TINY` web server.* Suppose a `TINY` web server (described in CSAPP:11.6) is going to be hosted on 10.0.0.32, at port 8000. However, a careless admin modifies line 5 of `TINY`'s `read_requesthdrs` (Figure 11.32) to be "`unsafe_readlineb(rp, buf, MAXLINE);`". The code of `unsafe_readlineb` is shown below.

```
ssize_t unsafe_readlineb(rio_t *rp, void *usrbuf, size_t maxlen)
{
    int n, rc;
    char c, *bufp = usrbuf;

    n = 1;
    while (1) {
        if ((rc = rio_read(rp, &c, 1)) == 1) {
            *bufp++ = c;
            if (c == ’\n’) {
                n++;
                break;
            }
        } else if (rc == 0) {
            if (n == 1)
                return 0;
            else
                break;
        } else
            return -1;
        n++;
    }

    *bufp = ’\0’;
    return n-1;
}
```

   Since `unsafe_readlineb` does not check `maxlen`, a buffer overflow (Section 3.10.3) may happen. Now suppose you are an attacker and want to kill the web server so that the others can no longer access the website. You somehow know 3 key info about the remote process ("`./tiny 8000`"):

   1. the "`_exit`" function is at `0x7ffff78a9dd0`.
   2. when execute `read_requesthdrs`, the RBP is `0x7ffffffec020`.
   3. when execute `read_requesthdrs`, the variable `buf` is at `0x7ffffffea020`.

   How do you construct a HTTP request to do the attack?

2. We have learned *HTTP (hypertext transfer protocol)* in *CSAPP:11.5*. Like HTTP, *SMTP (Simple Mail Transfer Protocol)* is also an application-level protocol. It is widely used for our daily email transmissions.

   (a) The domain name of the SMTP server of SJTU is (`smtp.sjtu.edu.cn`). What is the IP address of this server? List at least 2 methods to get the IP address from its domain name.

(b) The server is listening on port *25*. Use "`telnet smtp.sjtu.edu.cn 25`" to setup a connection to the server.

(c) Unlike HTTP requests that have a header+body layout, SMTP requests are *commands*. Type the command "`EHLO` *`<your hostname>`*" to greet the server.

(d) Then login with the command "`AUTH LOGIN`". You are prompted for a username then a password. Note that both the prompt and your answer are base64-coded. For example, to decode the prompt, use "`echo -n "VXNlcm5hbWU6" | base64 --decode`". To encode your answer, use "`echo -n "username or password" | base64`". After this, you should see the response "Authentication successful" from the server.

(e) Then you can compose a email using the commands "`MAIL FROM`", "`RCPT TO`", "`DATA`". For example, to send an email from A@sjtu.edu.cn to B@163.com, you can type (the ending "`<CR><LF>.<CR><LF>`" is used to end the "`DATA`" content)

```
MAIL FROM: A@sjtu.edu.cn
RCPT TO: B@163.com
DATA

From: Alice A@sjtu.edu.cn
To: Bob B@163.com
Subject: It works!

Hi,

This is from Alice.

.
```

(f) What would happen if the "`MAIL FROM`" is still A@sjtu.edu.cn while the "From:" in the "`DATA`" is "Curry curry@nba.org"?

3. Which level would the following data being shared? Answer with *not shared*, *threads*, or *processes*. For example, if X is shared between threads but not shared between processes, answer *threads*.

File descriptor table	______
File table	______
Stack	______
Heap	______
Program counter	______
Condition code	______
Installed handler	______
V-node table	______