Hw2

Code Review:

1. This code has a great OO design. It has a proxy class, a HTTP request class, a HTTP response class and a cache class, which all have clear roles. They do their related jobs separately. And all classes and their methods are well commented. The readability and abstraction are good. No duplicate codes.

2. This code doesn’t provide a basic/strong exception guarantee. It doesn’t use try/catch throughout the code and sometimes the proxy daemon will be killed by some requests silently.

3. This code doesn’t use a docker to run.

4. This code applies RAII.

Functionality:

Achieved:

1. The proxy can achieve required Methods of GET, POST and CONNECT though it cannot access some of the websites.

2. The proxy can cache responses. And it follows the rules of expiration time and re-validation.

3. The proxy utilized concurrency. And it handles synchronization using mutex.

4. The proxy has a log file to record requests.

5. The code is in good C++ style.

Not-achieved:

1. The proxy is not robust to external failures. It will be killed sometimes.

2. The proxy makes no exception guarantees.

3. The proxy doesn’t set up docker for submissions.

4. No testcases provided.

Possible Attacks:

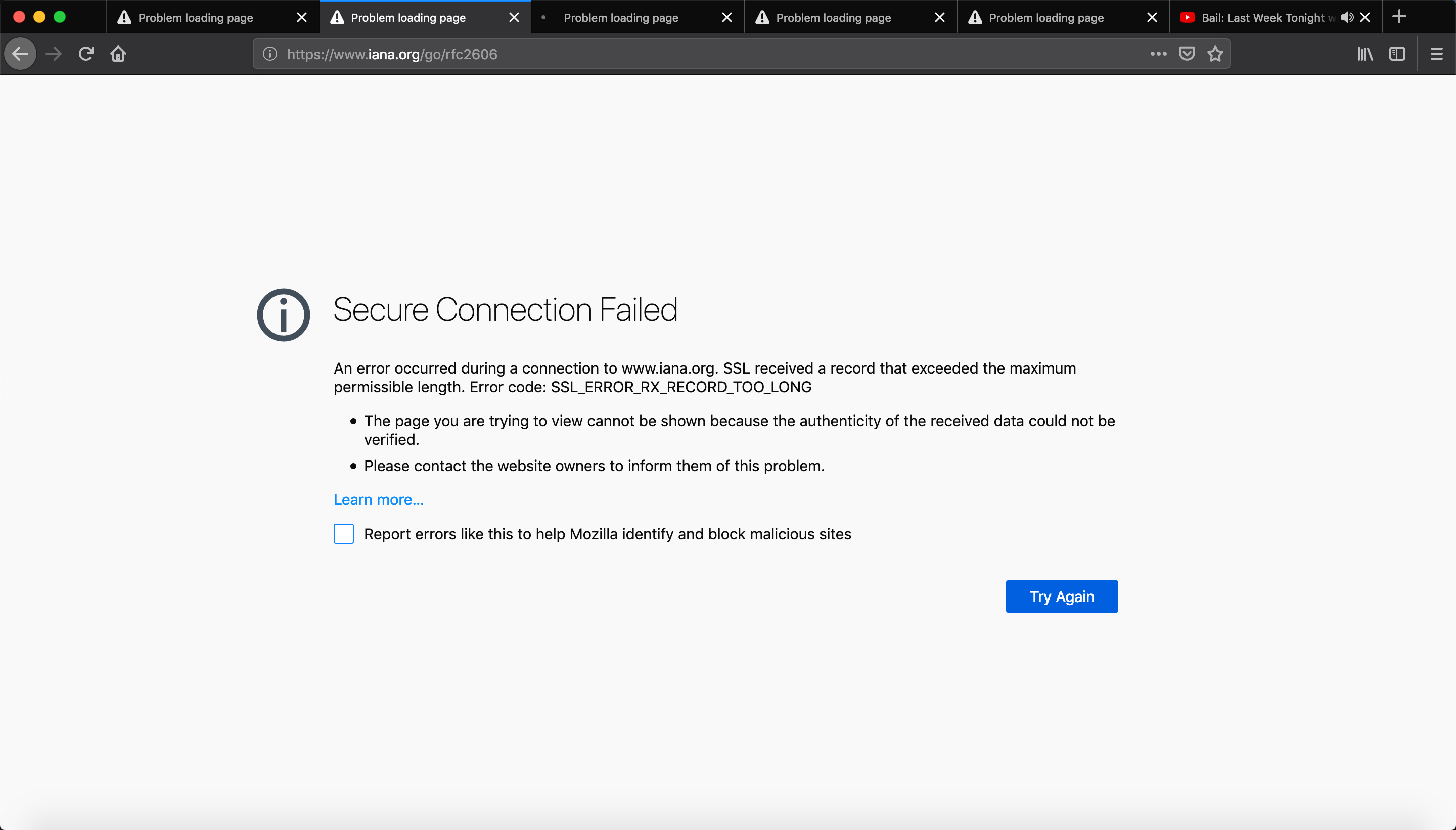
1. Sometimes the daemon will be killed after some requests. This means the proxy has a no guarantee on exception and it cannot handle errors.

2. Cannot open [www.sohu.com](http://www.sohu.com), which has a lot of contents.

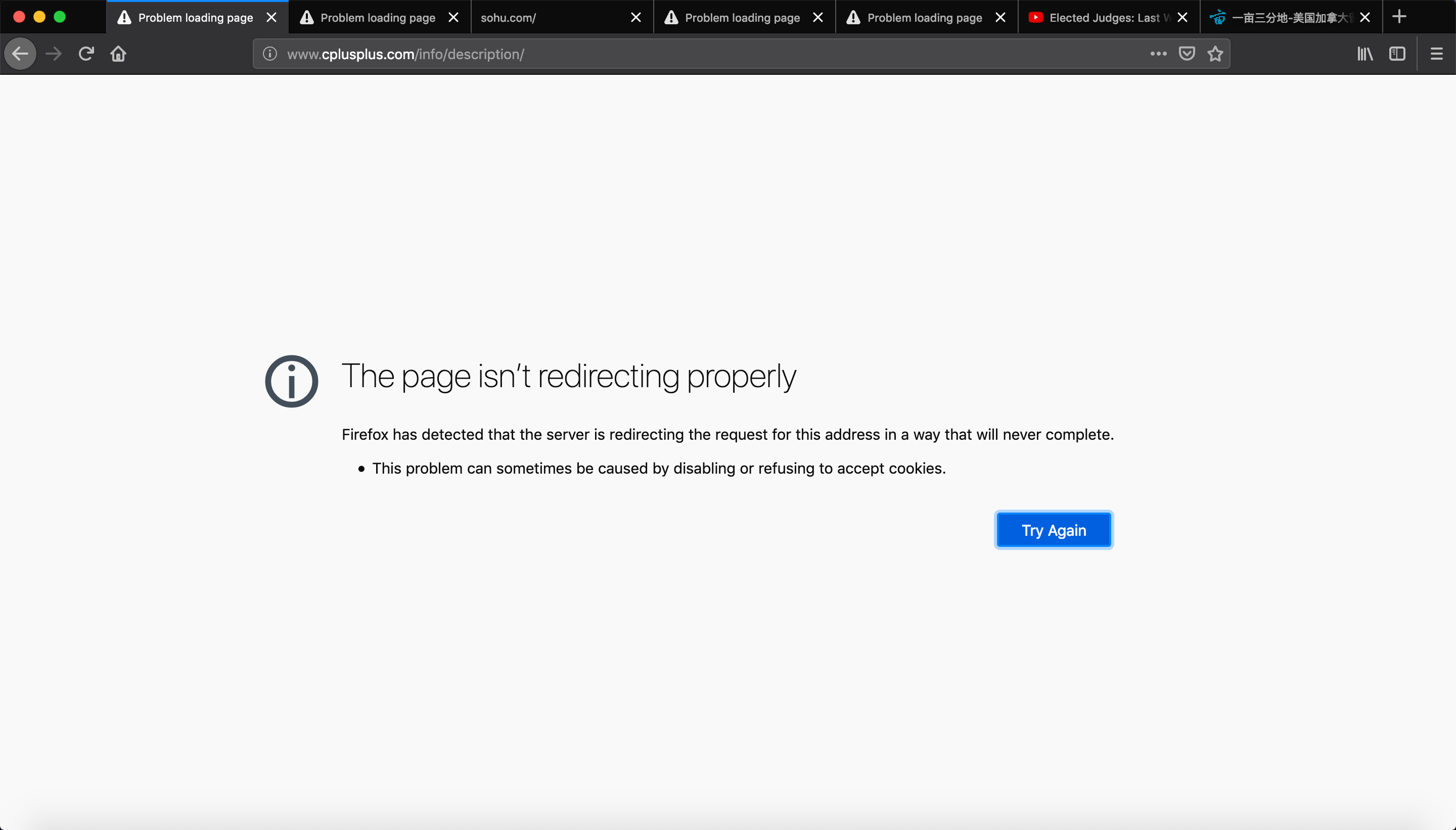
3. Different requests have the same unique id. Which should be also different.



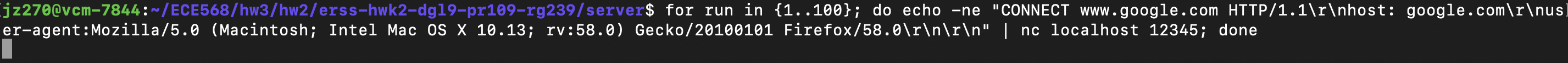
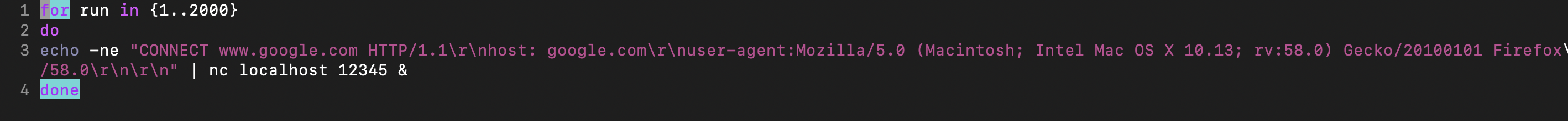
4. When we try to access the following website, it says the authenticity of the received data could not be verified. There may be something wrong with the request sent and there may be security problems.

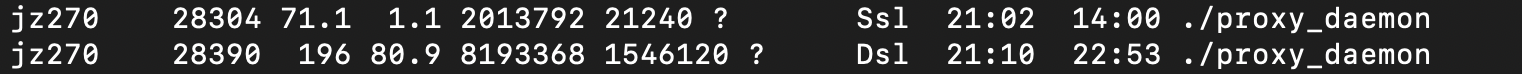


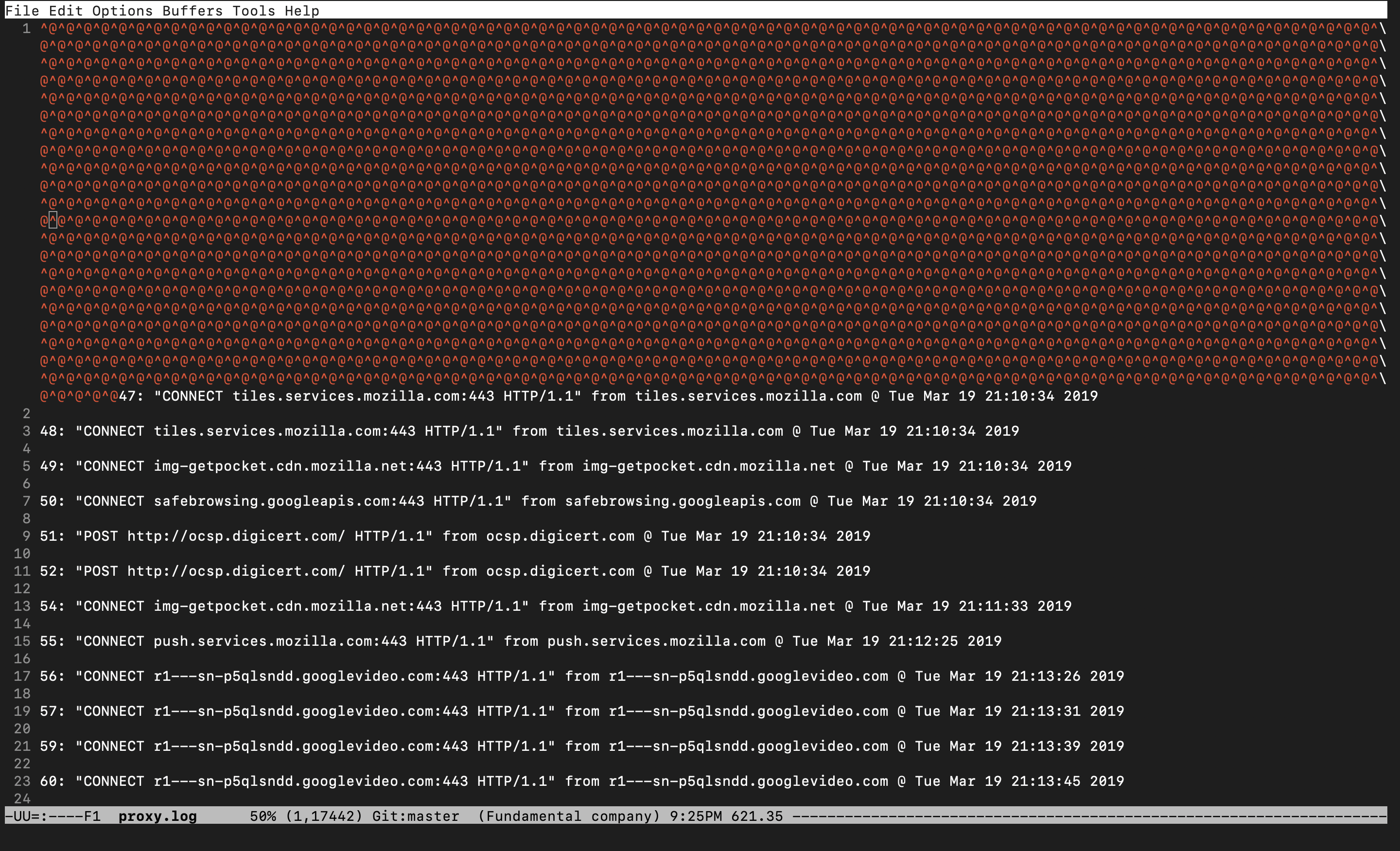
5. When we try to access the link on www.cplusplus.com, it says the page isn’t redirecting properly. It also kills the proxy daemon.



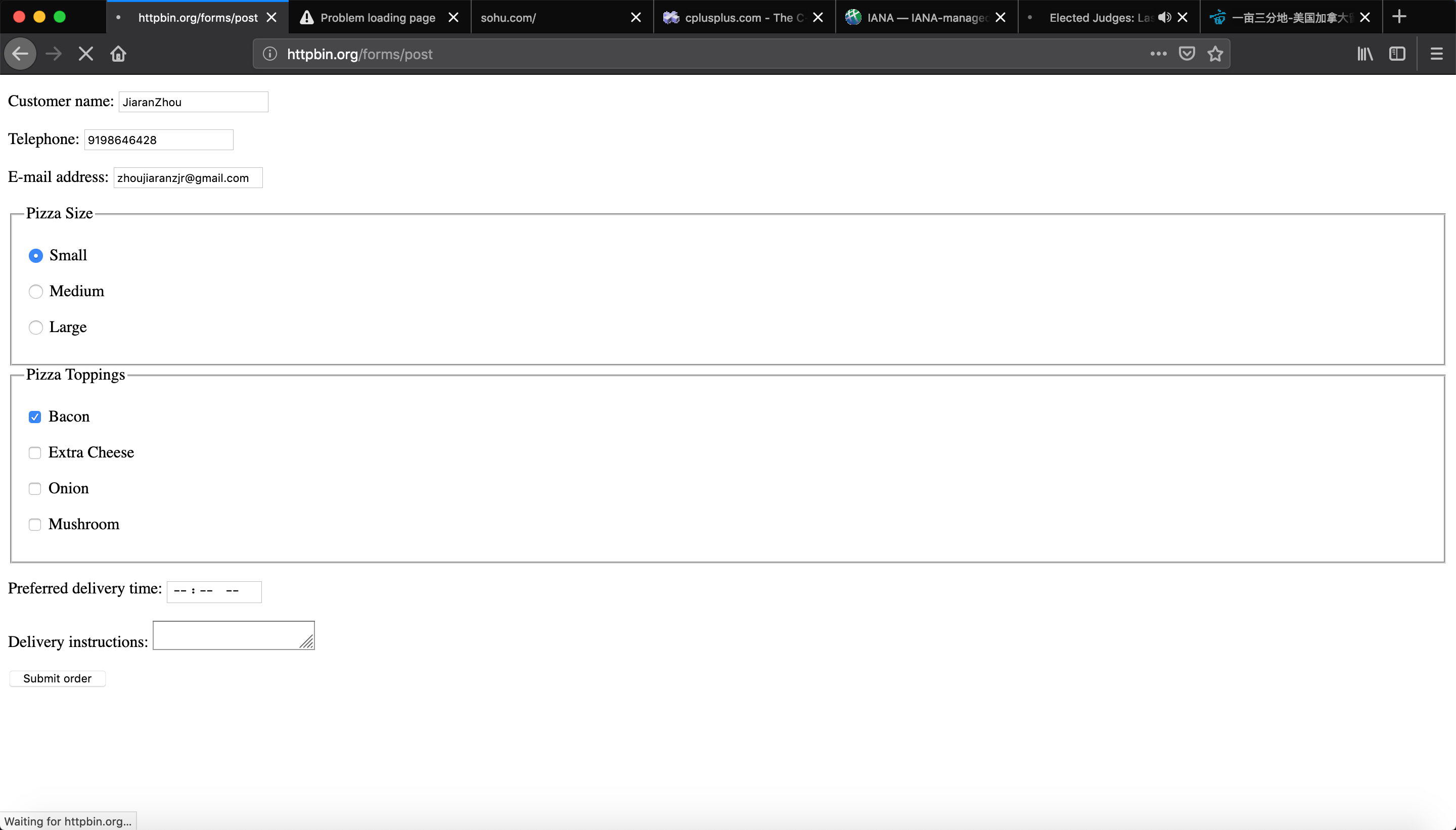
6. We tried to use netcat to run CONNECT [www.google.com](http://www.google.com) for 100 times. The command is as follow. But it will make my virtual machine very slow. We use ps -aux to see what happens. We can see there are two daemon process, which means its parent didn’t exit properly. And although there is no buffer overflow, they occupy 71.1 and 196 percent CPU. When we looked at its log, it’s like the following figure. There must be something wrong with their log.







7. When we tried the POST Method using the website provided by the professor, it cannot successfully post. There must be something wrong with the POST Method.



8. The proxy cannot open chunked website using GET Method. We tested it using the following website:

http://www.httpwatch.com/httpgallery/chunked/chunkedimage.aspx