# Lab 2 Questions

1. **As part of requirements you have been asked not to store the users’ passwords in plain text. What security risks can such an action cause? How can this problem be addressed programmatically?**

This is very bad. If the server is hacked and the database compromised, then the attackers can steal the users identities, possibly also on other sites since many users reuse their passwords. Also, in a bigger organization, you might not want your employees to have access to the users passwords in plain text since that also is a security risk. This can be addressed by hashing the passwords with a secure algorithm like MD5. A randomly generated and secure unique salt should also be prepended to every password before hashing to avoid the risk of dictionary attacks/rainbow tables.

1. **As http requests and responses are text­based information, they can be easily intercepted and read by a third­party on the Internet. Please explain how this problem has been solved in real­world scenarios.**

The most common solution is to use a secure data transmission protocol, especially when transmitting high-risk data, such as passwords and credit card information. The most popular one that is also considered to be very secure is HTTPS that essentially is an extension of HTTP with SSL encryption. HTTPS uses both public and private keys in its certification procedure to establish a handshake between the client and the server. A session key is generated that is only valid for a short period and time, meaning that an attacker would not have long time access even if the private key somehow is intercepted.

1. **Do you think the Telnet client is a good tool for testing server­side procedures? What are its possible shortages?**

I do not think it is a good tool. For one, all information is transmitted in plain text that means that it can be easily intercepted. Since I don’t hash passwords on the client side, this would be extremely bad in a real life scenario. It is also kind of cumbersome to use, especially if you type all HTTP headers manually. What is good with telnet is that it is quite easy to set up and get some fast testing done, especially if you are testing locally. It is also a good way to learn the HTTP protocol and because of that a good educational tool.