PART 1

Exercise 1.1

Consider a signature scheme in which a message is hashed using a collision-resistant 64-bits long hash function. Describe an attack that succeeds with high probability in 2^{32} steps.

Exercise 1.2

What is an asymmetric key certificate and why is it important? What is the risk if we accept an unverified (or self signed) certificate during a SSL/TLS connection to a web server? Does it make sense, from a security point of view, to store the insecure SSL/TLS certificate for future connections?

Exercise 1.3

Define the one-time-pad cipher and show that it is a perfect cipher. Assume that a Man-In-The-Middle attacker knows the structure of the sent message and wants to modify a part of it (for example a specific value in a certain point of the message). Is this possible? Illustrate through a simple example.

PART 2

Exercise 2.1

Consider the following fragment of a C program:

```
...
read_name() {
    char name[64];
    printf("name: ");
    scanf("%s",name);
    printf("Hello %s! How are you?\n", name);
    ...
}
```

Explain how it can be exploited to jump into an arbitrary memory location. Discuss mitigations performed by operating systems and show how to fix the program.

Exercise 2.2

What is Netfilter? Show a simple configuration of Netfilter that drops all traffic except ssh connections (use pseudo-iptables syntax or Mignis if you prefer).

Exercise 2.3

Consider the following PHP fragment

```
<?php
    ...
    echo "Username: " . $_GET['username'];
    ...
?>
```

Discuss possible attacks and show how to fix the code so to prevent them.