Exercise 1 (10 points)

Program `login` checks user's password provided on the command line. It exhibits the following behaviour:

```
$ ./login AAAAAAAAAAAA
ACCESS DENIED!
$ ./login AAAAAAAAAAAAAAAA
ACCESS DENIED!
$ ./login AAAAAAAAAAAAAAAAAA
*** stack smashing detected ***: ./login terminated
```

1. What kind of bug is present in the program? Illustrate through a fragment of source code (3 points)
2. Explain in detail the enabled security mechanism (3 points)
3. What attack would be possible if the security mechanism were disabled? Show a simple example of exploitation (2 points)
4. Illustrate how the security mechanism prevents your proposed exploitation (2 points)

Exercise 2 (10 points)

Consider the following firewall configuration of host `myhost.com`:

```
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:http
ACCEPT     all  --  anywhere             anywhere            state ESTABLISHED
Chain FORWARD (policy ACCEPT)
Chain OUTPUT (policy ACCEPT)
```

Discuss which of the following connections are enabled and which are instead blocked. Motivate your answers and explain the role of the "`state ESTABLISHED`" rule.

1. Outgoing connections to `http://www.google.com` (2 points)
2. Incoming connection to `http://myhost.com` (2 points)
3. Incoming connection to `https://anotherhost.com` (2 points)
4. Outgoing `ssh` connections to `myprovider.com` (2 points)
5. Incoming `ssh` connections to `myhost.com` (2 points)

Exercise 3 (10 points)

Consider the following PHP fragment

```
<?php
    ...
    $query = "SELECT 1 FROM people WHERE email = '" . $_POST['email']. "'";
    ...
?>
```

If the query is successful (i.e., at least one row is returned) the application goes on asking for more user input, otherwise message ``email does not exist'' is displayed.

1. What kind of vulnerability is present? (3 points)
2. Explain how an attacker can discover the vulnerability (give the attack payload) (2 points)
3. Even if the web application does not directly leak the content of the database, indirect leakage is possible. Give a high level description of the attack steps (it is not necessary to show the actual SQL code) (3 points)
4. Illustrate a security mechanism that prevents the previous attack (2 points)