Instituto Superior Técnico  
Universidade de Lisboa

**Smart Restaurant**

Network and Computer Security, Alameda, Group 1

|  |  |  |
| --- | --- | --- |
| 77900 | Luís Filipe Abreu dos Santos | [Foto] |
| 77961 | Pedro Filipe Ferreira Fernandes | [Foto] |
| 77966 | Catarina Alexandra Pereira Cepeda | [Foto] |

# 1. Problem

Given the restaurant scenario we have # key issues to address in terms of security. The first being client privacy, we need to guarantee that any personal information like account and orders information cannot be accessed by unauthorized parties. Another issue is ensuring that all orders can be properly to a given individual which is a classical non-repudiation problem.

(Given the chosen scenario, why is security necessary? What is the main problem being solved? Use around 200 words)

# 2. Requirements

1. **Client**

* Client privacy while ordering (decoupling orders from the clients);
* Protection of user accounts and payment information (including order records if these are to be kept);
* Authenticity of the restaurant (client application must be able to authenticate the restaurant);

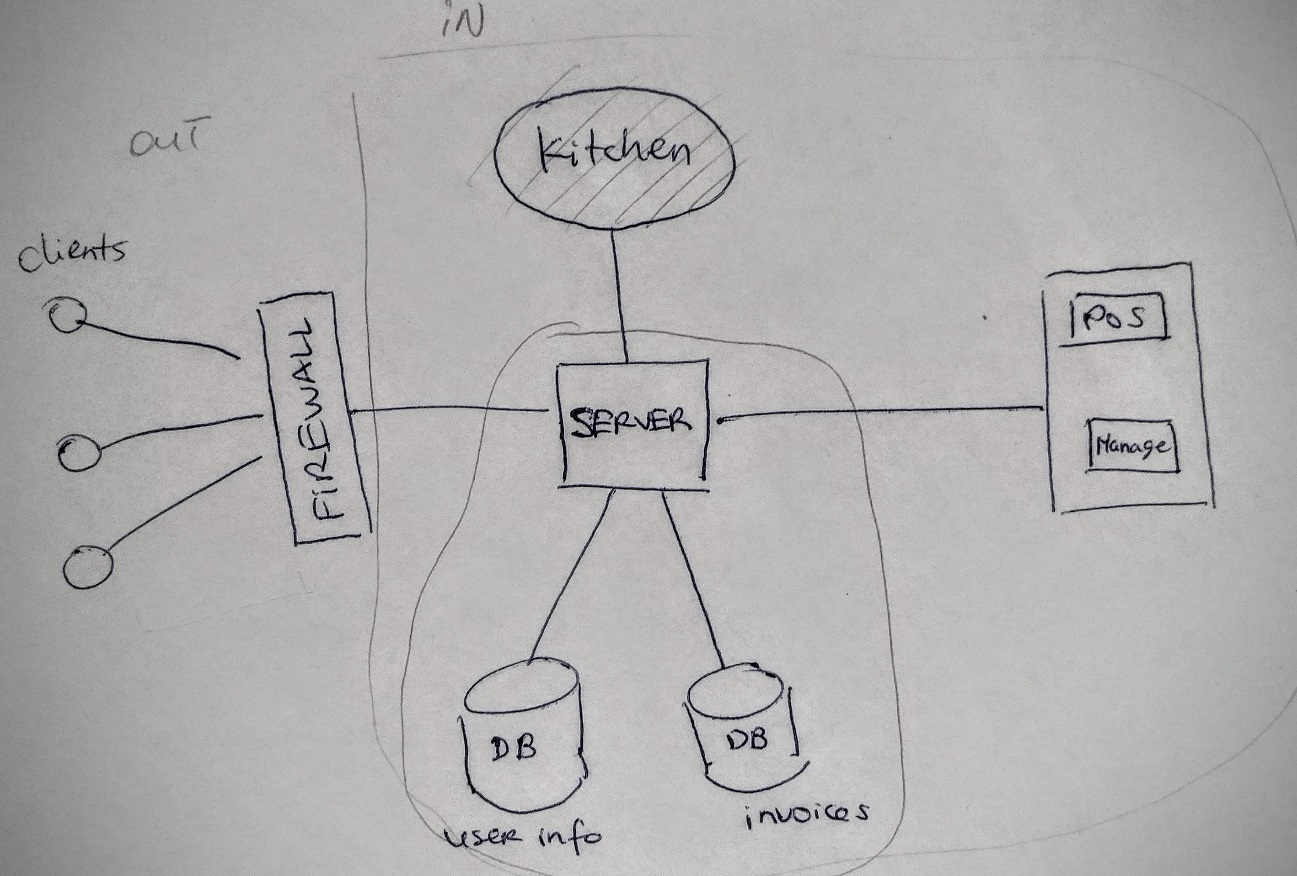
1. **Restaurant**

* Locality of the client while ordering;
* Authenticity of incoming orders from the client;
* Non-repudiation of orders therefore preventing the restaurant from having orders placed that won’t be paid for;
* Separate the public and private networks and data flow between them;
* (Application level) Firewall for incoming requests;
* Ensure menu information sent to the client is correct;

Key threats:

* Fake clients ordering food;
* Access to private information;
* Fake payment requests from attacker;

# 3. Proposed Solution



The above diagram highlights key components of the proposed architecture for the system. We will assume that all key-pairs have been previously installed and distributed between the internal components and that the external clients use an authentication protocol like token passing or Oauth2.

Basic:

Ensure the authenticity of all connected components and make sure that any messages sent are clearly identified and confidential in the case of communication between the clients and the Server.

Intermediate:

Separate data of incoming requests so that personal information cannot be retrieved unless by a trusted party (ie Manager). Ensure that the information in the user database is confidential.

Isolate the internal network and implement a application level firewall for incoming requests from the outside.

Advanced:

Protect payment transactions??? K-anonimity

# 4. Tool Preferences

Android Studio – Installed

Restful API – Angular maybe???

MySQL Database? – Installed

Log4j – tested

found/installed/tested/well-tested

# 5. Work Plan

|  |  |  |  |
| --- | --- | --- | --- |
| **Week** | Catarina | Luis | Pedro |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

(Table containing one row per week until the submission date; and one column per group member with expected activities for the given week; some cells may be blank because of other courses. State clearly when basic, intermediate and advanced versions are expected to be achieved)