Uma imagem com pessoa, homem, interior

Descrição gerada automaticamenteUma imagem com texto, pessoa, homem, interior

Descrição gerada automaticamente

Uma imagem com pessoa, homem

Descrição gerada automaticamente

**RESTAURANTS & TOURISM: THE CORK SIRS – TAGUSPARK - GROUP T48**

Bernardo Castiço ist196845

Hugo Rita ist196870

Pedro Pereira ist196905

# Build Infrastructure

## I. Business context

O tema do projeto é construir um serviço seguro denominado TheCork que permita aos seus usuários reservarem mesas para refeições nos restaurantes que planeiem ir comer.

**II. Infrastructure overview**

VM1 dos clientes (Java): Client\_T48 | IP: 192.168.0.100/24 que liga à Firewall\_T48.

VM2 que funciona como Firewall: Firewall\_T48 | IP: 192.168.0.10/24 para ligar à VM1 Client\_T48 & IP: 192.168.1.254/24 para ligar à VM3

WebServerCork\_T48 & IP: 192.168.2.254 para ligar à Database\_T48.

VM3 que corre a aplicação (Backend -> Java | Frontend -> JavaScript): WebServerCork\_T48 | IP: 192.168.1.1/24 que liga à Firewall\_T48.

VM4 da base de dados (PostgreSQL): Database\_T48 | IP: 192.168.2.4/24 que liga à Firewall\_T48.

VM5 que corresponde ao banco (Java): Bank\_T48 | IP:192.168.0.1/24 que liga à Firewall\_T48.

Na página em baixo temos uma imagem que mostra a nossa infraestrutura visualmente, no entanto, vamos também explicá-la agora por palavras.

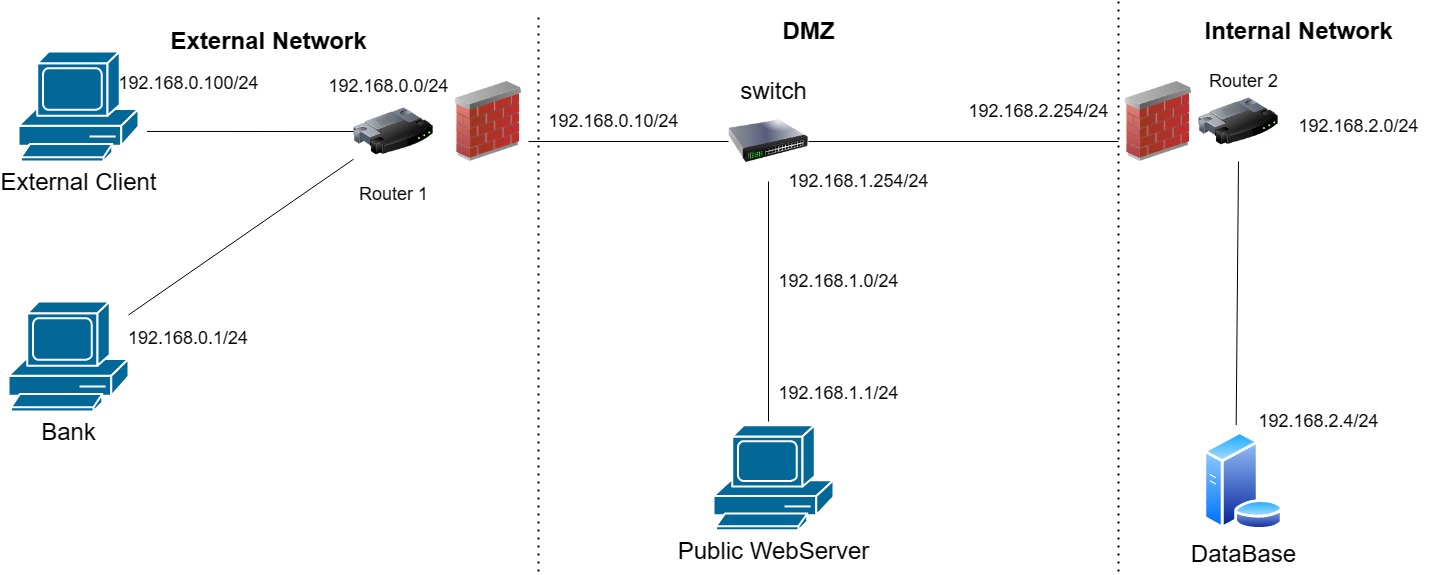
A virtual machine responsável pelos pedidos dos clientes tem o nome de Client\_T48 e a virtual machine da Firewall que se chama Firewall\_T48 estão ligadas na mesma network através do sw-1.

Por sua vez, a virtual machine WebServerCork\_T48 e a virtual machine Firewall\_T48 também estão ligadas na mesma network, mas desta vez através do sw-2.

A virtual machine Database\_T48 e a virtual machine Firewall\_T48 também estão ligadas na mesma network, mas desta vez através do sw-3.

Por fim, a virtual machine Bank\_T48 que é responsável pela comunicação entre o banco e TheCork e está ligada à virtual machine Firewall\_T48 pelo sw-1.

A aplicação permite ao utilizador usufruir das seguintes 4 operações sobre as suas reservas: Create, Read, Update e Delete.



Firewall rules:

* A firewall apenas aceita pedidos ssh (port22) e http (port80) vindos da VM Client\_T48.
* Todas as conexões http (port80) da VM Client\_T48 são redirecionadas para a VM WebServerCork\_T48.
* Todas as conexões ssh (port22) da VM Client\_T48 são redirecionadas para a VM Database\_T48.
* Pedidos da VM Database\_T48 só são aceites se forem ssh

(port22).

* A VM WebServerCork\_T48 só pode começar conexões com a

Internal Machine.

* A Internal Machine não pode começar conexões.
* A VM Bank\_T48 só pode começar conexões com a VM WebServerCork\_T48.

## III. Secure Communications

*What existing security protocol is being used?*

O protocolo de segurança que decidimos usar é o TLS. Escolhemos este protocolo porque este ser um protocolo usado mundialmente para garantir privacidade e segurança nas comunicações através da internet.

As ligações que configuradas com TLS são as seguintes: Cliente – webServer e Banco – webServer.

As mensagens trocadas encontram-se encriptadas pelo algoritmo RSA.

*Who is communicating?*

O cliente e o webServer comunicam entre si, o banco e o webServer também comunicam entre si e o mesmo acontece entre a base de dados e o webServer. São todos estes canais que pretendemos proteger usando o protocolo TLS.

*What keys exist and how are they distributed?*

O banco tem um par de chaves pública/privada.

O webServer tem um par de chaves pública/privada.

O cliente o banco e o webServer têm todos acesso às chaves públicas uns dos outros.

O cliente e o banco validam a chave pública do webServer através do certificado digital emitido pela CA que associa a entidade webServer a essa mesma chave pública. O cliente e o banco podem validar a assinatura do certificado usando a chave pública da CA e se a CA for de confiança e a assinatura for válida então o cliente e o banco podem confiar na chave pública.

Recorrendo ao mesmo mecanismo o webServer valida a chave pública do banco.

## IV. Security Challenge

(ii) TheCork’s users keep their credit card data information stored in the app in order to facilitate reservations at high-end restaurants that need you to leave a deposit. This data needs to be safe and confidential at the device level and on-the-wire when it gets sent to TheCork’s servers and to the Credit Card company.

*What is the main problem being solved?*

O problema principal a ser resolvido é garantir a autenticidade dos utilizadores, confidencialidade da conexão entre os “endpoints” e a integridade da informação partilhada.

*What security requirements were identified for the solution?*

R1. A confidencialidade das conexões entre o cliente e o webServer e o webServer e o banco tem de ser garantida.

R2. Os utilizadores do TheCork devem ser autenticados quando usam o sistema.

R3. As mensagens trocadas nas conexões entre o cliente e o webServer e o webServer e o banco devem ter a garantia de que não serão alteradas respeitando o princípio de integridade.

R4. A frescura nos canais Cliente – webServer e Banco – webServer tem de ser garantida.

**V. Proposed solution**

*Who will be fully trusted, partially trusted, or untrusted?*

Na rede interna todas as conexões são fully-trusted, isto é, o webServer e a DataBase são fully-trusted.

No início da comunicação, todos os pedidos da rede externa são untrusted, após o “handshake”, um user passa a ser partially-trusted.

A rede usada pelo cliente e pelo banco é unstrusted.

Os colaboradores são fully-trusted.

*How powerful is the attacker? What can he do and not do?*

O atacante vai tentar fazer replay attacks. Também irá tentar ter uma posição passiva ou mesmo ativa num ataque man in the middle, tentando alterar os dados que são passados nas mensagens.

O atacante também tentará fazer SQL/PHP Injection através de pedidos enviados por ele.

O atacante poderá tentar personificar clientes, caso descubra as suas credenciais da conta, obtendo assim toda a informação sobre a conta desse utilizador, dado que tem total controlo sobre a conta do mesmo.

*Who will comunicate and What keys will exist and how will they be distributed?*

O cliente e o webServer vão comunicar entre si. O banco e o webServer também vão comunicar entre si, assim como o webServer e a DataBase também vão comunicar entre si.

Nas mensagens enviadas pelo cliente para o webServer, uma vez que não é realista o cliente ter acesso a uma CA para assinar a sua chave pública, decidimos optar pela opção de existir um PreMaster Secret entre o cliente e o webServer. Este será encriptado com a chave pública do webServer e enviado para este onde o próprio o irá decifrar com a sua chave privada. Posto isto, tanto o cliente como o webServer geram com base nesse número uma session key.

As comunicações começadas pelo webServer e que têm como destino o banco serão cifradas pelo webServer com a chave pública do banco e o banco irá decifrá-las com a sua chave privada. Como estamos a usar cifra de chave publica/privada as chaves publicas são conhecidas por todos. Por sua vez, as chaves privadas são conhecidas por quem as detém.

O cliente e o banco validam a chave pública do webServer através do certificado digital emitido pela CA que associa a entidade webServer a essa mesma chave pública. O cliente e o banco podem validar a assinatura do certificado usando a chave pública da CA e se a CA for de confiança e a assinatura for válida então o cliente e o banco podem confiar na chave pública.

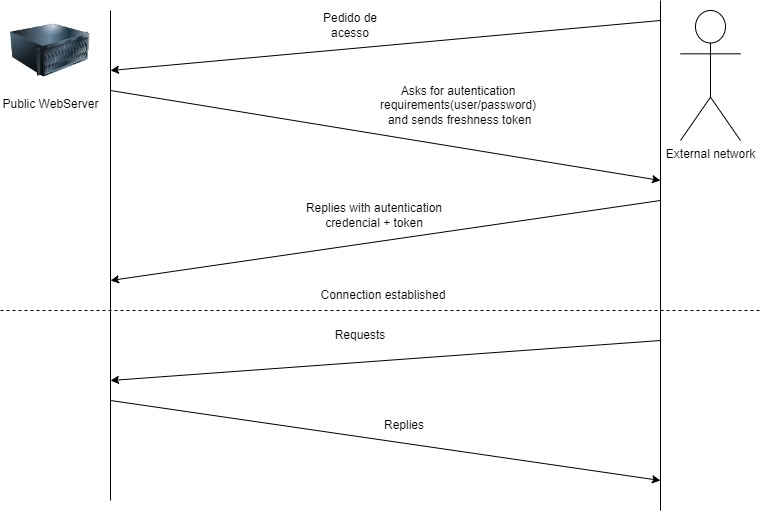
Recorrendo ao mesmo mecanismo o webServer valida a chave pública do banco.

*Which* *security properties will be protected?*

O nosso protocolo assegura a confidencialidade, integridade e frescura da informação enviada pelo cliente através da app móvel e pelo website e vice-versa.

O nosso protocolo assegura a confidencialidade, integridade e frescura da informação enviada pela app móvel e pelo site para o banco e vice-versa.

O nosso protocolo assegura a autenticidade dos clientes quer no site quer na app móvel.

Sequence Diagram: