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 onde abre o site TheCork: 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 e VM5 Bank_t48 & IP: 192.168.1.254/24 para ligar à VM3 WebServerCork_T48 & IP: 192.168.2.254 para ligar à Database_T48.
- VM3 onde corre Backend & API -> Java & Spring Boot e o 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 onde corre o servidor do banco (Java): Bank_T48 | IP:192.168.0.200/24 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.

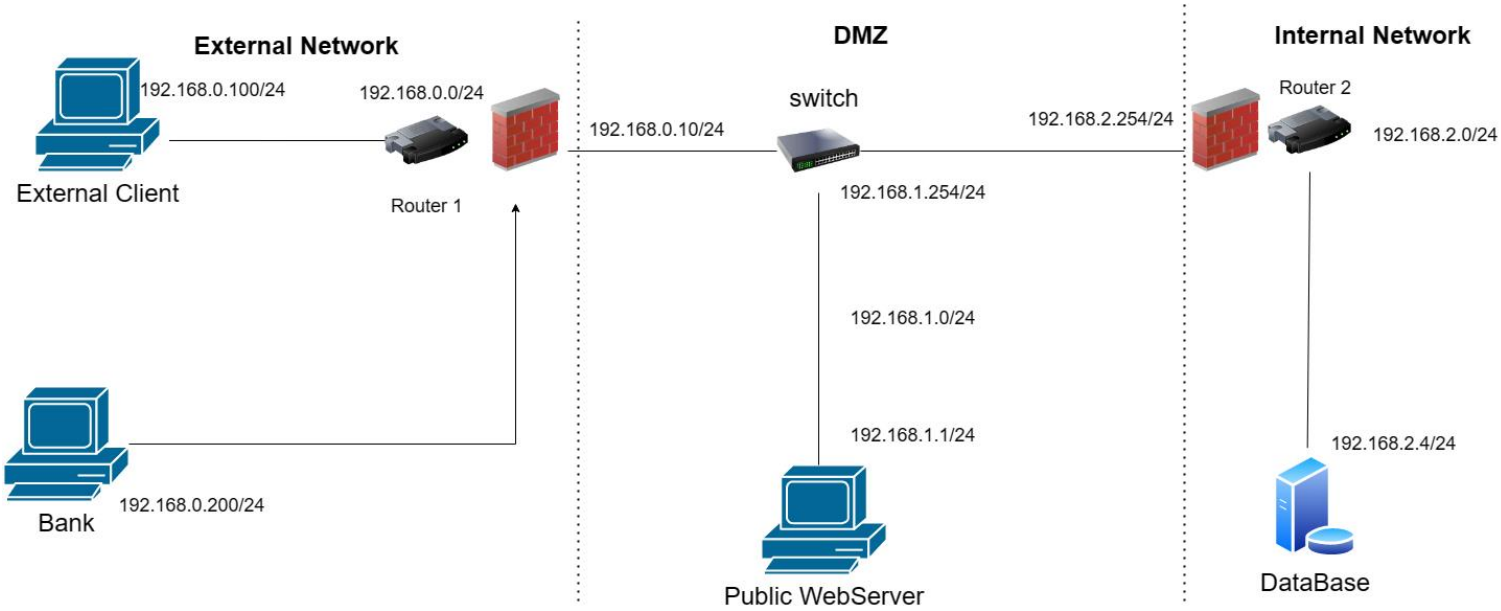
A virtual machine responsável pelos pedidos dos clientes tem o nome de Client_T48 e a virtual machine da Firewall que se chama Firewall_T48 estão ligadas na mesma network através do sw-1.

A virtual machine responsável por correr o servidor do banco tem o nome de Bank_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.

A aplicação permite ao utilizador reservar mesas e as reservas podem ser verificadas na Database.



Firewall rules:

- Todas as conexões HTTP (port80) da VM Client_T48 são redirecionadas para a VM WebServerCork_T48.
- Todos os pedidos começados pela VM Database_T48 são rejeitados.
- A VM Bank_T48 só pode começar conexões com a VM WebServerCork_T48.
- Pedidos da VM WebServerCork_T48 para a VM Database_T48 são aceites.
- Pedidos no port 5432 da VM WebServerCork_T48 são redirecionados para a VM Database_T48.

III. Secure Communications

What existing security protocol is being used?

Os protocolos de segurança que decidimos usar são o TLS e o HTTPS que iremos especificar mais abaixo. Escolhemos estes protocolos por serem protocolos usados mundialmente para garantir privacidade e segurança nas comunicações através da internet.

As ligações que configuradas com HTTPS é a ligação Cliente – webServer e com TLS Banco – webServer.

Who is communicating?

O cliente comunica com o webServer através do site do TheCork comunicam entre si, o banco e o webServer também comunicam entre si e o mesmo acontece entre a base de dados e o webServer.

O canal webServer - banco é protegido com TLS, por sua vez o canal cliente – webServer é protegido com HTTPS e por fim o canal webServer – Database não é protegido uma vez que assumimos que a rede interna é seguro.

What keys exist and how are they distributed?

Numa solução ideal o banco teria um par de chaves pública/privada.

O webServer tem um par de chaves pública/privada.

O banco tem acesso à chave pública do webServer e no cenário ideal o webServer também teria acesso à chave pública do banco se este a tivesse.

A chave pública do webServer é certificada através do certificado digital emitido por uma CA fictícia.la CA que associa a entidade webServer a essa mesma chave pública.

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 tem de ser garantida.

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

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

R4. A frescura no canal Banco – webServer tem de ser garantida.

R5. A confidencialidade das conexões entre o banco e o 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 estabelecidos os protocolos de segurança as ligações entre a rede externa e interna passam a ser partially-trusted.

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 quando o cliente aceder ao site do TheCork e esta ligação encontrar-se-á protegida por HTTPS.

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

O banco ao estabelecer uma ligação para para o webServer gera um número random (preMaster Secret) que o banco envia para 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 banco como o webServer geram com base nesse número uma session key utilizando o algoritmo SHA3-256. Ao enviar informação para o banco o server, o server cifra a informação necessário com a session key e o banco usa também a mesma session key para cifrar a informação e enviar para o webServer.

Posto isto, de modo a verificar a integridade fazemos um digest da informação cifrada (payload) com o algoritmo SHA3-256 e ciframos o resultado deste digest (hmac) com a session key. Quando a informação chegar ao outro lado da conexão, deciframos o hmac com a session key que é partilhada pelos 2 lados da conexão e em seguida fazemos o digest da informação (payload) e comparamos com o hmac recebido.

Isto aplica-se tanto à informação enviada do webServer para o banco como à informação do banco para o webServer.

De modo a garantir a frescura das mensagens quando a conexão é estabelecida o webServer gera um token que é um inteiro qualquer e envia este na mensagem inicial de confirmação da conexão para o banco.

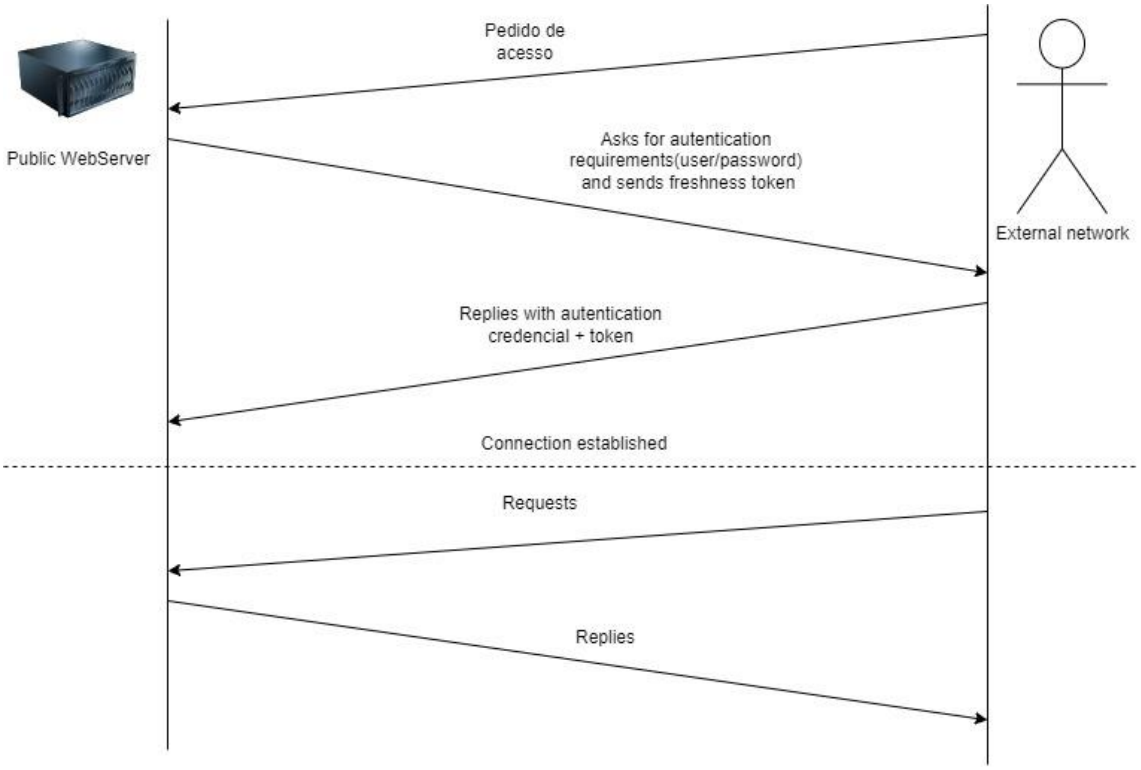
Which security properties will be protected?

O nosso protocolo assegura a confidencialidade e integridade 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:



VI. Results

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

Este requisito é assegurado ao usarmos o protocolo HTTPS na comunicação entre o cliente e o webServer.

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

Este requisito é assegurado uma vez que quando os clientes para entrarem no site TheCork têm de se autenticar com credenciais já existentes na base de dados. Para testar esta funcionalidade criados previamente na database 3 registos e para se aceder ao site terão de ser colocadas as credenciais correspondentes às do registo. Assumimos ainda que o login segue o processo de autenticação de 2 fatores e que existe um sistema de autenticação onde quando se faz um refresh de página e se muda de página as credenciais já colocadas aquando do login são de novo autenticadas.

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

Este requisito é assegurado graças ao protocolo já explicado em cima e que se segue em seguida:

De modo a verificar a integridade fazemos um digest da informação cifrada (payload) com o algoritmo SHA3-256 e ciframos o resultado deste digest (hmac) com a session key. Quando a informação chegar ao outro lado da conexão, deciframos o hmac com a session key que é partilhada pelos 2 lados da conexão e em seguida fazemos o digest da informação (payload) e comparamos com o hmac recebido.

Isto aplica-se tanto à informação enviada do webServer para o banco como à informação do banco para o webServer.

R4. A frescura no canal Banco – webServer tem de ser garantida.

Este requisito é assegurado graças ao protocolo já explicado em cima e que se segue em seguida:

De modo a garantir a frescura das mensagens quando a conexão é estabelecida o webServer gera um token que é um inteiro qualquer e envia este na mensagem inicial de confirmação da conexão para o banco.

R5. A confidencialidade das conexões entre o banco e o webServer tem de ser garantida.

Este requisito é assegurado graças ao protocolo já explicado em cima e que se segue em seguida:

Nas mensagens enviadas pelo banco para o webServer decidimos optar pela opção de existir um PreMaster Secret entre o banco 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 banco como o webServer geram com base nesse número uma session key. Ao enviar informação para o banco o server, o server cifra a informação necessário com a session key e o banco usa também a mesma session key para cifrar a informação e enviar para o webServer.

Lista de assumptions necessárias para o bom funcionamento da nossa solução:

- A autenticação é suportada pelo mecanismo autenticação de 2 fatores (SMS para o telemóvel).

- Existe um sistema de autenticação onde quando se faz um refresh de página e se muda de página as credenciais já colocadas aquando do login são de novo autenticadas.

VII. Conclusion

Observações

No código submetido a API não tem o código necessário para correr em HTTPS uma vez que o browser não aceita o import do certificado usado para certificar esta ligação e desta forma não deixaria prosseguir o funcionamente correto do site.

Como tal, removemos a parte do código responsável por HTTPS. Para que este protocolo se encontre ativo devem-se seguir os seguintes passos.

No ficheiro applications.properties situado em SIRS_PROJECT/TheCorkApi/src/main/resources devem-se adicionar no início as seguintes linhas de código:

```
server.ssl.key-store-type=PKCS12

server.ssl.key-
store=classpath:keystore/samplekey.p12
server.ssl.key-store-password=thecork

server.ssl.key-alias=samplekey
server.ssl.enabled=true
```

Também tem de se mudar o código nos ficheiros da diretoria SIRS_PROJECT/frontend_sirs/src/pages colocando “https” em vez de “http” nas chamadas à API (LoginPage.js linha: 21 | RestaurantDetails.js linha: 40, 73)

Na nossa comunicação entre o banco e o webServer ciframos as mensagens utilizando sempre a session key gerada na comunicação entre o banco e o webServer. Na situação ideal a informação individual a enviar (card number, 3digitsCode, validityDate) teria que ser cifrada com uma session key diferente a cada envio, e esta chave secreta teria de ser cifrada com a chave pública do banco e ambas estas informações enviadas para o banco com o protocolo usado.

Um ataque possível para o qual não nos conseguimos defender é um DoS attack. Isto deve-se à limitação das Virtual Machines, dado que a VM onde corre o webServer tende a ficar lenta ao correr o frontend e o backend ao mesmo tempo.

Com o protocolo de segurança que delineámos e implementámos conseguimos criar um site TheCork seguro que comunica com um banco externo e uma base de dados.

Ao TheCork é possível ligarem-se vários clientes de modo a usufruírem do serviço que este disponibiliza e assim deliciarem-se com refeições dos seus restaurantes favoritos.