

**Instituto Superior de Engenharia**

**de Lisboa**

**Engenharia Informática e de Computadores**

***SEGURANÇA INFORMÁTICA***

***2ºSÉRIE***

Docente: José Simão

Filipe Fé nº 42141

Inês Gomes nº 42160

José Cunha nº 43526

**Índice**

**Exercício 1** ..................................................................................................................................... 2

[**1.1** 4](#_Toc6200)

[**1.2** 7](#_Toc6201)

[**1.3** 7](#_Toc6202)

[**Exercício 2** 8](#_Toc6203)

**Exercício 1**

**1.1)**

Estamos perante o caso em que tanto o cliente como o servidor se querem autenticar, ou seja, existe uma autenticação mútua. No contexto do protocolo TLS, os materiais criptográficos necessários que têm de ser configurados no cliente:

* Um conjunto de certificados raiz de forma a verificar se o certificado enviado pelo servidor, que contém a sua chave pública, é válido.
* O seu próprio certificado que contém a sua chave pública
* A sua chave privada para realizar uma assinatura que comprova a sua autenticidade

**1.2)**

No processo de handshake, após o servidor enviar para o cliente o seu certificado (que contém a sua chave pública), o cliente irá gerar um pre master secret que será cifrado utilizando a chave pública recebida, caso o certificado recebido seja válido. Esse pre master secret cifrado irá ser enviada ao servidor (desafio) de modo a verificar que este possui a sua chave privada.

O servidor obtém o pre master secret descodificando a mensagem recebida, utilizando a sua chave privada.

Após este processo, este será usado como chave no esquema simétrico de autenticação MAC , de forma a provar que, de facto, possui a chave privada.

Depois de ser estabelecida uma chave segura (pre master secret) a ser usado por ambos é garantido que o canal de comunicação é fiável.

**1.3)**

O record protocol é responsável pela transferência de blocos de dados entre os dois intervenientes na comunicação. Através dos parâmetros negociados no handshake, é gerada uma marca de autenticidade (através do MAC) e os dados são encriptados para futuramente serem enviados ao cliente ou ao servidor.

O problema é que um atacante pode dedicar-se a tentar descobrir o padding válido usado nas mensagens encriptadas por um desses intervenientes, como é descrito em ataques baseados no de Vaudenay. Para esse efeito basta guardar as mensagens que foram trocadas entre o cliente e o servidor e mais tarde usar a técnica descrita. Desta forma consegue obter a chave usada no processo de cifra. Caso o servidor tenha essa chave do seu lado e esta seja de longa duração, neste momento é possível ao atacante decifrar as mensagens que são enviadas nesse canal de comunicação (até a chave ser alterada).

**Exercício 2**

**2.1)**

O cliente/relying party indica os recursos a que pretende ter acesso através de um pedido HTTP, cujo URL contém um query parameter ***scope*** com essa informação.

**2.2)**

A framework OAuth 2.0 é usada para garantir autorização no acesso a determinados recursos.

Ao fazer uso dele para autenticação, não existe um processo de asserção sobre o utilizador que se pretende autenticar, pois apenas é gerado um access token que garante acesso temporário aos recursos. Ao usar esse token para autenticação não existe a garantia de que é o mesmo utilizador responsável pelo seu pedido.

**Exercício 3**

**3.1)**

O id token (fornecido pelo Identity-Provider à aplicação web) tem o propósito de informar a aplicação web de que o cliente se autenticou.

**3.2)**

A aplicação cliente.