

**2ª Série de Exercícios**

**Licenciatura em Engenharia Informática e de Computadores**

## Segurança Informática

# *Prof. Eng. José Manuel Simão*

# *Grupo 7:*

*Hugo Fora 42121*

*Luana Silva 42189*

*Rui Lima 42200*

*Bruno Lourenço 42400*

*Lisboa, 24 de novembro de 2017*

1. 1. Canal inseguro. Autenticidade é garantida comparando os *MAC's* das mensagens anteriormente enviadas.
   2. A autoridade a ser comprometida é a autoridade que produz os certificados raiz em que o cliente confia, pois se o atacante conseguir que a autoridade emita um certificado válido para o atacante usar então o cliente ao ver esse certificado como confia na autoridade iria estar a comunicar com o atacante e não com a autoridade.
   3. O uso de chaves públicas e privadas para a troca do *master secret* entre cliente e servidor é considerado obsoleto/inseguro pelos browsers, pois se o atacante descobrir a chave privada do servidor consegue alcançar os *master secrets* usados nas ligações anteriores e consequentemente decifrar as mensagens destas comunicações.
   4. Não, pois as *cookies* estão guardadas no browser e qualquer utilizador com acesso à máquina que corre o browser consegue ver a estrutura das *cookies*.
2. Se não for usado o fator aleatório *salt* então com um ataque de dicionário pode ser descoberto a *password* mais facilmente pois os valores de *hash* podem ser calculados anteriormente.
3. O *client\_secret* só é usado em comunicacões directas entre o *client* e o *authorization server* pois só é necessário para autenticar o cliente (App web) em relação ao *relying party*, para isso teve que haver um registo do cliente na *relying party*.
4. 1. Após o utilizador introduzir as credenciais no formulário do servidor de autenticação e este for validado, o servidor de autenticação responde com um *302 redirect* para a App web com o código resultante da validação do utilizador.
   2. Não é obrigatório o *relying party* validar o *id\_token* quando o *id\_token* é apenas usado pelo o *relying party* uma vez que este confia no *identity provider* e sabe que o valor é válido.