Respostas

1 –

a) No *record protocol* a autenticidade nas mensagens é garantida pela utilização do MAC (*Message Authentication Codes*).

b) A deteção de inserção ou adulteração maliciosa da mensagem é realizada do seguinte modo:
    1. Após a definição dos algoritmos de encriptação, o cliente gera um sequência confidencial (*pre master secret*) cifrada com a chave pública do servidor.
    2. Numa fase seguinte, com a sua chave privada, o servidor irá decifrar essa sequência gerada pelo cliente, após a qual será calculada o algoritmo HMAC da mesma juntamente com todas as mensagens enviadas anteriormente.
    3. Após receber o HMAC do servidor, o cliente poderá confirmar a sua integridade calculando o algoritmo HMAC por si próprio e comparar com a enviada pelo servidor.
    4. Caso o resultado calculado pelo cliente for diferente do calculado e enviado pelo servidor, conclui-se que houve alguma inserção/adulteração das mensagens.

c) O estabelecimento da *pre master secret* apenas através da chaves pública e privada não garante o *perfect foward secrecy* uma vez que todo o processo é dependente da confidencialidade da chave privada do servidor, caso esta seja comprometida, é possível que o atacante aceda às chaves e aos IVs o que, por conseguinte, caso tenha acesso as sessões de *handshake* anteriores conseguirá decifrar as mensagens.

2 – Num cenário em que o atacante tenha acesso ao *hash* do utilizador e ao respetivo *salt*, o processo será totalmente dependente da robustez da palavra-passe e o número de tentativas será um fator inafetivo. Ou seja, neste caso especificamente, há duas possíveis vertentes:

1. Caso a palavra-passe do cliente seja robusta o suficiente para não estar presente no dicionário o ataque será inútil.
2. Caso a palavra-passe esteja no dicionário, é certo que o atacante através de uma procura exaustiva chegue a palavra-passe comparando o hash deste cliente com o hash de cada palavra-passe do dicionário juntamente com o *salt* deste mesmo utilizador, o que levará, consequentemente, a necessidade de apenas uma única tentativa no processo de autenticação.

3 –

a) Uma vez que o atacante conhece o esquema do cookie e sabe do identificador do utilizador bem como a função de Hash, basta que este gere um cookie com as informações do atacando.

b) Uma das alterações possíveis para evitar este tipo de ataque é utilizando um esquema MAC, garantindo assim a confidencialidade do cookie, cifrando o seu conteúdo. E uma vez que o atacante não tem conhecimento da chave utilizada no esquema MAC, não consegue gerar um cookie para aquela servidor, mesmo tendo acesso às informações de um utilizador (o atacando).

4 –

a) O valor presente no scope é gerado pela aplicação cliente.

b) Há uma comunicação indireta entre o cliente e o servidor que são quando o cliente é redirecionado pelo browser para o servidor durante o pedido de autorização, e segundo quando o servidor envia a resposta a esse pedido de autorização para o browser, e o browser reencaminha o cliente através do endereço de callback.

c) A diferença entre os dois itens mencionados reside no facto de que o id_token contém a informação base sobre o utilizador autenticado que pode ser acedida pelo cliente, enquanto o access_token contém a informação base mais informação adicional que deverá apenas ser acedida pelos servidores de recursos, nunca acedida pelo cliente.

5 –

a) O princípio de privilégio mínimo está relacionado com a família RBAC no sentido em que esta tem um conjunto de características entre os seus membros que envolvem uma hierarquia de papéis (roles) e restrições para cada utilizador.
Cada um dos membros da família RBAC tem características diferentes.

O modelo RBAC0 implementa a relação entre user assigment (que contém as características do utilizador) e permission assigment (que contém um conjunto de permissões para a sessão), servindo de base para os restantes membros.

O modelo RBAC1 implementa o conjunto de hierarquias de roles na qual o utilizador escolhe qual o role que quer ativar, herdando os roles juniores do mesmo (as permissões são as diretamente associadas ao role do utilizador mais as dos roles júnior).

O modelo RBAC2 utiliza as restrições como um mecanismo para impor regras de organização que podem ser aplicadas às relações user assigment e permission assigment (separação de deveres).

O último modelo da família RBAC, RBAC3, implementa a hierarquia de roles do RBAC 1 e as restrições do RBAC2 (num cenário onde a administração é delegada a terceiros, pode ser necessário impor restrições).

b) Uma vez que pc pertence a r4 e este é sénior de r2, é impossível que r2 tenha acesso a pc, sendo este júnior de r4. No entanto é possível que r2 tenha acesso a pb, já que r2 é sénior de r1 e este tem acesso a pb.