47233 – Nuno Bartolomeu

47235 – Pedro Lemos

48306 – Raul Santos

Grupo 1

1.

a)

A autenticidade nas mensagens no record protocol é garantida através da marca MAC que é gerada neste sub protocolo a partir da chave (K), que é do conhecimento do cliente e do servidor, em junção com a informação em si.

b)

O handshake deteta a inserção ou adulteração maliciosa de mensagens através do Master Secret que é gerado utilizando 2 valores aleatórios, 1 do Cliente e 1 do Servidor em junção com parâmetros que foram concordados por ambos.

c)

Na versão do TLS em que o pre master secret é estabelecido usando chaves públicas e privadas, não é garantida a propriedade perfect forward secrecy porque a partir do momento em que a chave privada é comprometida seria possível descobrir o pre master secret de conexões passadas utilizando essa chave. Isto não é o caso no Diffie-Hellman uma vez que o pre master secret é criado através de números aleatórios.

2.

Este erro facilita um ataque de dicionário através da interface de autenticação onde o número de tentativas é limitado, porque, para entrar pela interface só é necessário saber o nome de utilizador e a password respetiva. O nome de utilizador já é conhecido e a password pode ser descoberta usando um programa auxiliar que usa um dicionário de passwords, aplica o metodo de hash a cada uma delas em conjunto com o salt do utilizador e compara o hash final com o hash do utilizador. Como é díficil ter H(x) = H(y), quando for encontrado um hash igual a probabilidade de a password estar certa é muito elevada.

3.

a)

Um atacante pode fazer-se passar por outro utilizador, para o qual sabe o seu identificador, utilizando um cross-site request forgery attack, onde o utilizador entra no site do atacante que guarda a informação do cookie e depois utiliza-a para entrar na aplicação web com as credenciais do utilizador.

b)

Para evitar o ataque anterior pode ser alterada a flag same-site para o valor strict, isto faz com que o cookie não possa ser enviado para sites de atacantes.

4.

a)

O valor indicado no scope é definido pelo dono de Recursos, mas tem de ser colocado corretamente pela aplicação cliente

b)

O cliente e o servidor de autorização comunicam indiretamente através do browser do dono de recursos, sempre que o cliente enviar um pedido para o dono de recursos incluindo um Access token no header de autenticação.

c)

O access token é a representação das credenciais necessárias para o cliente poder aceder a certos recursos do servidor, sendo enviado sempre que necessário para o dono de Recursos para autenticação no Header de autenticação. O id token é um conjunto de informação sobre um utilizador autenticado, este token é assinado pelo fornecedor de identidade usando JSON Web Token.

5.

a)

A familia de modelos RBAC contribui para a implementação do princípio de privilégio mínimo devido à possibilidade de organização de roles (garantida em RBAC1 e RBAC3) que, dependendo do esquema implementado, podem garantir a utilização de privilégios minimos para executar a operação em questão.

b)

Verificando a hierarquia dos roles apresentada na figura, podemos concluir que o utilizador u2 não tem acesso ao recuso R, pois, apesar de ter a permissão pb devido ao seu role R2, necessita da permissão pc que é só garantida pelo role R4 que u2 não tem.

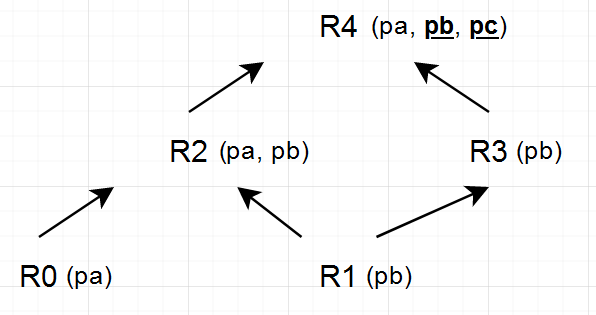


Figura1 – Representação do RBAC do exercício, com as permissões que cada role tem

6.

a)

Para executar o servidor que não precisa de certificado corremos o programa nodejs “https-server-without-certificate-request.js”, e para existir necessidade de o utilizador colocar o certificado corremos o programa nodejs “https-server-with-certificate-request.js”.

Foi também necessário modificar o ficheiro host para incluir que o endereço [www.secure-server.edu](http://www.secure-server.edu) utilizasse o endereço do localhost.

b)

Não foi feito.

7.

Realizamos este exercício com sucesso. A política usada resume-se na definição de permissões de escrita/leitura para diferentes URIs, que depois é verificada num Middleware de verificação RBAC usando a biblioteca casbin, na definição de hierarquia de roles e na definição dos users. Como o nosso servidor só aceita pedido GET e POST, todos os pedidos GET são considerados como pedidos de leitura e todos os pedidos post são considerados como pedidos de escrita. Quando um utilizador não especificado na política entra, é-lhe dado automaticamente um role free no servidor (ver casbinHelper.js).