Segurança Informática - 2º Trabalho

Licenciatura em Engenharia Informática e de Computadores

44860	Inês Franco
44880	Ricardo Tomé
47179	Afonso Cabaço

1.1. Considere o protocolo TLS e as infraestruturas de chave pública:

(a) De que forma é garantida a autenticidade nas mensagens no record protocol?
O TLS garante a autenticidade das mensagens através da utilização do esquema MAC, que é gerado e juntado à mensagem antes de esta ser encriptada e enviada para o protocolo de transporte. Note-se que é gerado um esquema MAC diferente para cada direção do canal de transporte.

(b) O sub-protocolo handshake assume que o canal de comunicação não envia/recebe mensagem com confidencialidade nem integridade. Sendo assim, como é que o handshake deteta a inserção ou adulteração maliciosa de mensagens?
O sub-protocolo handshake é capaz de detectar interações maliciosas das mensagens através das mensagens Finished que garantem que ambos os endpoints recebam o mesmo conjunto de mensagens. As mensagens Finsished são sumários encriptados de todas as mensagens anteriores do sub-protocolo handshake. No final do protocolo se as mensagens de Finished do cliente e do servidor não forem iguais sabemos que houve adulteração e o protocolo acaba imediatamente. Como as mensagens Client Hello e o Server Hello possuem um número aleatório sabemos que todas as mensagens de handshake vão ser diferentes logo evitamos replay attacks.

(c) Considere a versão do TLS em que o pre-master secret é estabelecido usando chaves públicas e privadas. Porque motivo esta forma de estabelecimento de chaves não garante a propriedade perfect forward secrecy?

Na versão do TLS onde o pre-master secret é estabelecido utilizando chaves públicas e privadas não garante a propriedade perfect foward
secrecy, porque caso a chave privada do entrypoint seja adquirida pelo atacante este conseguirá decifrar todos os futuros handshakes entre
ambos os entrypoints, para além de se também tiver handshakes anteriores também será possível decifrar estes.

2. Considere uma aplicação web onde as passwords são armazenadas na forma hu = H(pwdu||saltu), sendo H um função de hash, pwdu a password do utilizador u e saltu um número aleatório gerado no momento do registo do utilizador u, em que || indica a concatenação de bits. Devido a um erro de programação, a informação sobre os utilizadores, hashs e respetivos salts, ficou exposta numa página da aplicação web. Discuta se este erro facilita um

ataque de dicionário através da interface de autenticação onde o número de tentativas é limitado.

Este erro de programação facilita um ataque de dicionário à aplicação web. Isto deve-se ao facto de, como o atacante tem acesso aos hashes e aos valores de salt. este pode tentar concatenar várias palavras prováveis com os salts de forma a chegar aos hashes e assim obter a palavra chave. Como também foram expostas as informações dos utilizadores também se torna mais fácil gerar palavras prováveis.

3. Considere uma aplicação web que guarda no browser cookies contendo o par (u, H(u)), sendo u o identificador de um utilizador e H uma função de hash. Assuma que a construção do cookie é conhecida. A comunicação entre browser e aplicação é feita sobre HTTPS.

(a) Como poderia um atacante fazer-se passar por outro utilizador para o qual sabe o seu identificador?

a) O atacante sabendo o identificador do utilizador e a forma como as cookies são criadas, assumindo que o atacante sabe como realizar
A função de Hash é gerar uma nova cookie através da concatenação entre a hash e o identificador, este poderá tentar criar uma nova cookie válida.

(b) Que alterações propõe para evitar o ataque anterior?

b) Este ataque pode ser evitado se em vez da cookie ser constituída por uma Hash, ser constituída por um método que garanta também a
autenticidade do utilizador, como HMAC.

4.

a) O valor indicado na scope é determinado pela aplicação cliente quando esta realiza o redirect para o endpoint de autorização.

b)

c) As principais diferenças entre o access_token e o id_token são:
-O id_token é suposto ser lido pela aplicação cliente, enquanto que no caso do access_token o cliente não consegue analisá-la;
-O id_token tem como formato uma JWT(Jason Web Token), enquanto que o access_token é uma sequência de strings;
-O id_token contém a informação do utilizador, enquanto que o access_token contém as permissões, dadas à aplicação cliente,
as quais definem o que esta pode aceder através do authorization server.

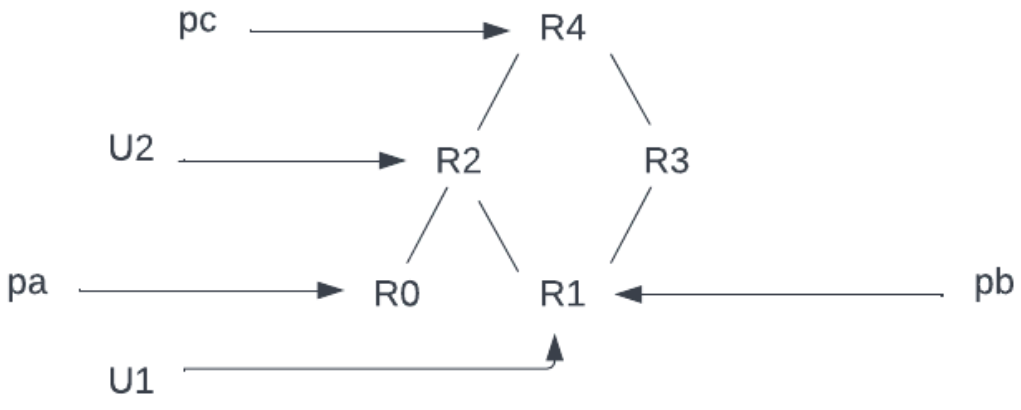
5. Considere o modelo de controlo de acessos RBAC1.

(a) Em segurança da informação, o princípio de privilégio mínimo determina que cada operação (ou conjunto de operações) deve ser realizada com o conjunto mínimo de permissões. De que forma a família de modelos RBAC contribui para a implementação deste princípio?
Nos modelos RBAC as permissões dos sujeitos estão encapsuladas nos papéis que estes sujeitos têm. Assim o sujeito apenas tem acesso a um subconjunto dos ficheiros disponíveis para o seu papel actual . Se o utilizador quiser aceder a outros ficheiros pode mudar o seu papel e assim aceder a outro subconjunto de outros recursos. Como os recursos ficam organizados por papéis dentro da organização, não é possível aceder a recursos fora deste papel e assim inviabilizar o princípio de privilégio mínimo.

(b)
Na política RBAC1:
U = {u1, u2}
R = {r0, r1, r2, r3, r4}
P = {pa, pb, pc}
UA = {(u1, r1),(u2, r2)}
P A = {(r0, pa),(r1, pb),(r4, pc)}
RH = {(r0 ≤ r2),(r1 ≤ r2),(r1 ≤ r3),(r2 ≤ r4),(r3 ≤ r4)}

Considere a existência da sessão s2, na qual está o utilizador u2. Neste contexto, o utilizador pretende aceder ao recurso W que exige a permissão pc, e pb. O utilizador u2 poderá aceder a este recurso?

O utilizador não pode aceder a este recurso. Como podemos observar na imagem abaixo o utilizador u2 apenas tem as permissões pa (herdada do role R0) e pb (herdada do role R1). A permissão pc apenas existe para o Role R4 que está acima na árvore e, logo, não está disponível para R2.



6. Configurações Realizadas

É necessário garantir a segurança entre a comunicação cliente browser.
Para isto são utilizados certificados. Por parte do servidor e também poderá ser usado pelo cliente

Inicialmente foram gerados três ficheiros PEM com a ferramenta OpenSSL

Ficheiro secure-server-cert.cer -> secure-server-cert.pem
Ficheiro secure-server-key.pfx -> secure-server-key.pem
Ficheiro CA1-int.cer -> CA1-int.pem

openssl x509 para ficheiros .cer
openssl pkcs12 para ficheiros .pfx

Depois foi feita a configuração mínima em options no servidor fornecido.
No browser para existir confiança entre o cliente e este servidor foi adicionado o certificado CA1 as raízes de confiança..
E no ficheiro secure-server-cert.pem foi adicionado o certificado intermédio presente em CA1-int.pem

Adição de uma linha no ficheiro hosts -> 127.0.0.1 www.secure-server.edu visto que o certificado só é aceito caso o domínio do url fosse esse.

Com estas configurações foi possível ligar ao servidor sem autenticação.

Para ser possível também com autenticação do cliente Alice_2.
Foram gerados mais dois ficheiros PEM.

Ficheiro CA2.cer -> CA2.pem

Foi realizada o resto da configuração de options, adicionando também o CA2 a certificate authority.

No browser foi adicionado também o CA2 a certificados raiz de confiança, assim como o CA2-int a certificados intermédios.
E adicionamos a *personal* o certificado, end-entity, da Alice_2 para conseguirmos testar se o protocolo https é estabelecido.

7. Modelo de Política

O Modelo RBAC - Este modelo funciona baseado em roles. Ou seja, é menos comum as permissões estarem associadas a x, y pessoa mas sim a uma role. Estas ditas roles são depois atribuídas às pessoas. É um bom modelo para organizações com grande dimensão, visto que não será necessário mudar permissões de várias pessoas que tenham esse papel, mas sim apenas a permissão da role a que essas pessoas estão associadas.

As permissões são assim atribuídas às roles e estas são sempre positivas. Um utilizador pode ter múltiplas roles e vai herdar as permissões que estão atribuídas às mesmas. Um utilizador pode também ter atribuído roles apenas num momento, isso é chamado de Session.

O Modelo RBAC1 é um modelo rbac0 mas com hierarquia de roles.
Roles com maior patamar, tem as permissões das roles inferiores a si. Existe uma relação de hierarquia entre elas.