Instituto Superior de Engenharia de Lisboa
Licenciatura em Engenharia Informática e de Computadores
Licenciatura em Engenharia Informática, Redes e Telecomunicações
Segurança Informática
Segunda série de exercícios, Semestre de Inverno de 19/20
Entregar até 25 de novembro de 2019

---

1. Considere o sub-protocolo *handshake* do TLS:

   1.1. Em que situação é usado um esquema de assinatura digital? Poderia ser substituído por um esquema MAC?

   1.2. No excerto de texto «[...] perfect forward secrecy prevents the recovery of information that was encrypted with older session keys», presente no RFC 7525 [1] sobre recomendações para uso seguro do TLS, quais são as chaves de sessão referidas e como poderiam ser obtidas pelo atacante?

2. O RFC 8018, *Password-Based Cryptography Specification*, especifica um algoritmo para transformar uma *password* numa chave simétrica [2]. Qual o papel do *salt* nesse processo?

3. Descreva como pode uma aplicação *web* garantir a autenticidade dos *cookies* que usa para manter estado de sessão, desde a geração à verificação.

4. No contexto da *framework* de autorização OAuth 2.0 :

   4.1. Qual o objectivo do parâmetro `scope`

   4.2. Admitindo que um atacante consegue ver toda a informação de e para o *browser* da vítima, é possível saber o `client_secret` de uma determinada aplicação web cliente? E o `client_id`?

   4.3. No contexto de um pedido de autorização desencadeado por uma aplicação cliente, depois do utilizador se autenticar no servidor de autorização e dar consenso, é enviado à aplicação cliente a identidade do utilizador?

5. No contexto do fluxo *authorization code* do protocolo OpenID Connect, para que serve o ID Token?

6. Pretende-se configurar e testar um servidor web com HTTPS, com e sem autenticação de cliente (*browser* e aplicação Java). Considere o certificado e chave privada do servidor `www.secure-server.edu` em anexo, o qual foi emitido pela `CA1-int` da primeira série.

   i) Configure e teste o servidor usando o cliente *browser*, com e sem autenticação de cliente. Tenha por base o ficheiro do servidor em anexo (`server.js`);

   ii) Realize a aplicação cliente Java;

   iii) Teste o servidor usando o cliente Java, com e sem autenticação de cliente;

   Descreva o material criptográfico que tem de configurar em cada situação e como o fez.

7. Realize uma aplicação Web com a seguinte funcionalidade:

   - A aplicação começa por autenticar os utilizadores. Os utilizadores são autenticados através do fornecedor de identidade social Google, usando o protocolo OpenID Connect [8];
   - Após autenticação do utilizador, a aplicação lista os *issues* do GitHub [4, 5] para um determinado projecto para o qual o utilizador tem acesso;
   - Os utilizadores autenticados podem criar *tasks* Google [7] a partir de *issues* do GitHub.

   Neste exercício não pode usar os SDK da Google/Github para realizar os pedidos aos serviços. Os mesmos têm de ser feitos através de pedidos HTTP construídos pela aplicação *web*.

   Considere os seguintes *endpoints* Google:

   - Registo de aplicações: `https://console.developers.google.com/apis/credentials`
   - *Authorization endpoint*: `https://accounts.google.com/o/oauth2/v2/auth`
   - *Token endpoint*: `https://oauth2.googleapis.com/token`
   - *UserInfo endpoint*: `https://openidconnect.googleapis.com/v1/userinfo`

e Github:

- Registo de aplicações: `https://developer.github.com/apps/building-oauth-apps/creating-an-oauth-app/`
- *Authorization endpoint*: `https://github.com/login/oauth/authorize`
- *Token endpoint*: `https://github.com/login/oauth/access_token`

Referências

[1] `https://tools.ietf.org/html/rfc7525`

[2] `https://tools.ietf.org/html/rfc8018`

[3] `https://en.wikipedia.org/wiki/Transport_Layer_Security#Client-authenticated_TLS_handshake`

[4] `https://developer.github.com/apps/building-oauth-apps/authorizing-oauth-apps/`

[5] `https://developer.github.com/v3/#authentication`

[6] `https://developer.github.com/v3/issues/`

[7] `https://developers.google.com/identity/protocols/OAuth2WebServer`

[8] `https://developers.google.com/identity/protocols/OpenIDConnect`

ISEL, 21 de outubro de 2019.