Práctica : PKI (2ª fase)

Objetivo

En la primera fase de esta práctica se pide crear una PKI simple donde la validación de certificados recae sobre un servidor OCSP; mediante el uso del comando `openssl` se actúa como cliente OCSP para solicitar la validación manual de los certificados.

En esta segunda fase, se configurará un sitio web seguro. El objetivo es comprobar la validez de los certificados involucrados en el acceso a dicho sitio web seguro. Para realizar esta validación se recurrirá al método de *Regular OCSP* y de *OCSP Stapling*[1].

Tareas

Se trabajará con la PKI creada en la primera fase para emitir los certificados necesarios.

Lanzar un servidor OCSP (*responder*) con `OpenSSL`

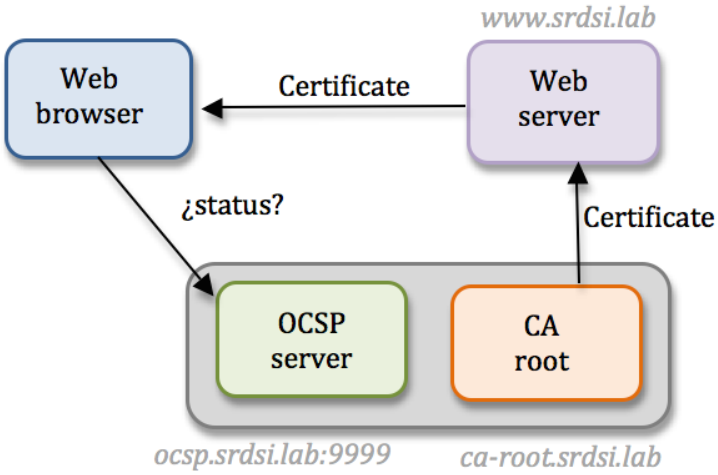
Para simplificar, en lugar de instalar/configurar un servidor OCSP, se pondrá en marcha un servidor OCSP con `OpenSSL`, de la misma forma que en la fase 1. Utilizará la base de datos creada por la CA, y esperará las peticiones de los clientes en el puerto indicado

```
[# rm log.txt ]

# openssl ocsp -port num_puerto -index indice_base_datos \
      -CA certificado-CA-root -rsigner certificado-ca-ocsp \
      -rkey clave-privada-ca-ocsp -text -out log.txt
```

[1] Disponible en Apache a partir de la versión 2.3

Regular OCSP



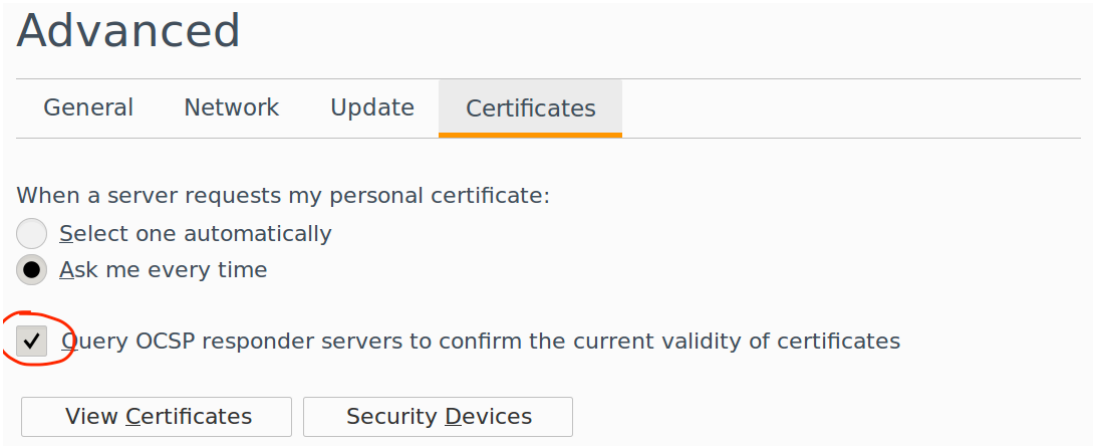
Crear sitios web en el servidor

- Sitio web seguro que presentará un certificado emitido por la PKI:
  `https://www.srdsi.lab:443`
- Sitios web asociados a la PKI
  - para la autoridad de certificación[2]
    `http://ca-root.srdsi.lab`
    que dejará acceso público a su certificado raíz y CRL
  - [para el servidor OCSP[3]:
    `http://ocsp.srdsi.lab:9999`]
    No es necesario crearlo, su función la asume el servidor OCSP lanzado con `openssl`

Configurar navegador del usuario

- Importar certificado del `ca-root`
- Comprobar que está activada la verificación mediante OCSP

[2] Como figure en el fichero de configuración de la PKI: `[default] aia_url`
[3] Como figure en el fichero de configuración de la PKI: `[default] ocsp_url`



Ejercicios

Probar accesos a `https://www.srdsi.lab:443`

- ¿Qué ocurre si no está en marcha el servidor OCSP? Comprobar con Wireshark[4] cuáles son los mensajes intercambiados.

  Antes los navegadores mostraban un mensaje de error, indicando que no podían contactar con el OCSP responder y no accedían a la página.
  Ahora los navegadores han cambiado este comportamiento y lo tratan como un ***`Soft-fail`*** *dando por válido el certificado presentado y mostrando la página.*
  Para verificar certificados proponen el uso de OCSP Stapling

- Poner en marcha el servidor OCSP y comprobar con Wireshark cuáles son los mensajes intercambiados[5] ¿Quién contacta con el servidor OCSP? ¿En qué momento contacta?

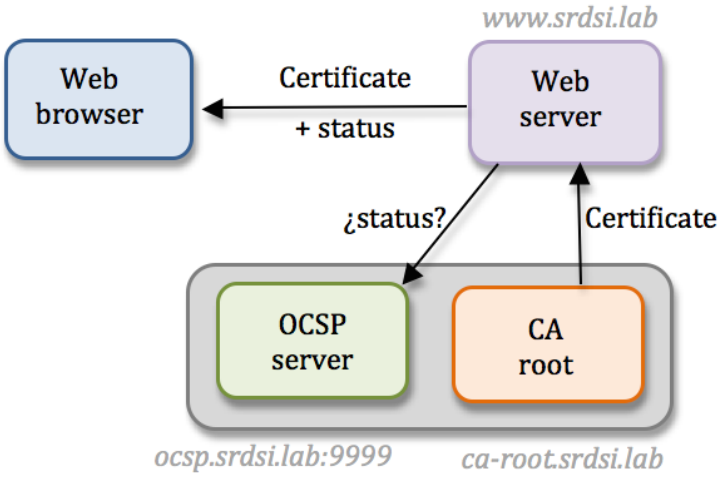
- Si revocamos el certificado del servidor web seguro[6] ¿Qué ocurre?

[4] Para visualizar las preguntas/respuestas OCSP, es necesario descodificar los segmentos TCP: `Decode As --> Transport --> HTTP`

[5] *Iceweasel* guarda durante un tiempo las respuestas de OCSP, es recomendable reiniciarlo antes de realizar una nueva captura.

[6] Reiniciar servidor OCSP, para que registre la revocación el certificado

OCSP stapling



Modificar sitios web

- Sitios web asociados a la PKI –> NO cambian
- Sitio web seguro que presentará un certificado emitido por la PKI:
  `https://www.srdsi.lab:443`
  - Activar el módulo `mod_socache_shmcb` [también `mod_ssl`]
    `$ a2enmod socache_shmcb`
  - En el fichero del sitio virtual añadir:
    Antes de `<VirtualHost>`
    `SSLStaplingCache "shmcb:logs/ssl_stapling(12000)"`
    Dentro de `<VirtualHost>`
    `SSLCACertificateFile` *`/pki/certs/ca-root.crt`*
    `SSLUseStapling On`
    `[ SSLStaplingForceURL http://`*`ocsp.srdsi.lab`*`:`*`9999`* `]`
    `[ SSLStaplingResponseMaxAge` *`90`* *# 1'5min*]

Ejercicios

Probar accesos a `https://www.srdsi.lab:443`

- Comprobar con Wireshark cuáles son los mensajes intercambiados ¿Quién contacta con el servidor OCSP? ¿En qué momento contacta?
- Si revocamos el certificado del servidor web seguro ¿Qué ocurre?