

Universidad del País Vasco

FACULTAD DE INFORMÁTICA

SEGURIDAD, RENDIMIENTO Y  
DISPONIBILIDAD DE SERVICIOS E  
INFRAESTRUCTURAS

*Práctica 2 PKI*

Autores:  
*Gorka Álvarez*  
*Jon Gámiz*  
*Asier Zubia*

Donostia, a 1 de Abril de 2021

## Índice

<b>1. Descripción del proyecto</b>	<b>2</b>
<b>2. Desarrollo de la práctica</b>	<b>2</b>
2.1. Regular OCSP	2
2.2. OCSP Stapling	4
2.3. Must-Staple	6

## Índice de figuras

1. Petición de certificado de acceso al sitio web	3
2. Captura <i>Wireshark</i>	3
3. Captura <i>Wireshark</i>	4
4. Captura <i>Wireshark</i>	5
5. Base de Datos de la CA raíz	5
6. Número de serie del certificado del servidor	5
7. Error de acceso por revocación del certificado del sitio web	6
8. Nueva directiva en la extensión del servidor	6
9. Comprobación de extensiones durante la firma del certificado del sitio web	7
10. Error de acceso debido a la indisponibilidad del servidor OCSP	7
11. Acceso correcto tras activar el OCSP	8

## 1. Descripción del proyecto

El objetivo de este proyecto es continuar con el desarrollo de la PKI<sup>1</sup> comenzada en la primera parte de esta práctica.

En la primera parte se creó una PKI simple donde la validación de certificados recaía sobre el servidor OSCP<sup>2</sup>. Mediante el comando *openssl* se actuaba como cliente OSCP para solicitar la validación manual de los certificados.

Durante 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 OSCP*, *OCSP Stapling* y *Must-Staple*.

## 2. Desarrollo de la práctica

Durante esta sección se describirá la creación y configuración de los distintos métodos de validación, *Regular OSCP*, *OCSP Stapling* y *Must-Staple*. Destacar que se ha utilizado el protocolo *TLS V 1.2* para poder visualizar los mensajes del protocolo *OCSP*.

Cabe destacar que para la primera parte de la práctica se había creado una PKI jerárquica, la cual estaba compuesta, a parte de por la CA raíz, por una CA subordinada. No obstante, debido a algunos problemas durante algunos apartados se ha decidido omitir la CA subordinada y tratar la CA raíz como única autoridad de certificación de la PKI.

Junto con este documento, se adjuntan los certificados, y claves, asociadas al sitio web, la CA raíz, el servidor OSCP y el de los usuarios que tratan de conectarse. Las contraseñas siguen el mismo formato que el establecido durante la práctica anterior, no obstante, para evitar inconvenientes, a continuación se listan los ficheros junto con sus respectivas contraseñas.

- **ocsp-server.key**. *ocsp-server*.
- **root-ca.key**. *root-ca*.
- **server.key**. *server*.
- **user1.key**. *user1*.

Atención: OSCP es independiente de CRL.

OCSP consulta la BD de la CA para comprobar el estado de los certificados, no la CRL.

### 2.1. Regular OSCP

El OSCP (*Online Certificate Status Protocol*) se encarga de verificar la revocación de un certificado frente a la CRL<sup>3</sup> y proporcionar una respuesta en tiempo real.

Antes de nada, se ha tenido que importar al navegador que se va a utilizar el fichero **.p12** que contiene el certificado y la clave asociada a ese certificado, del usuario, y el certificado de la CA raíz de nuestra PKI para el correcto funcionamiento del laboratorio.

En el enunciado no se menciona la necesidad de que el usuario presente un certificado

Primero se va a comprobar que ocurre cuando se intenta acceder al sitio seguro **https://grupo3-ca.grupotr3s:443** sin tener en marcha el servidor *OCSP*. Se ha podido observar que se pide escoger el certificado del usuario que se había importado previamente. En este caso, al haber importado un único fichero **.p12**, solo se puede hacer uso del mismo. Cabe destacar que se podría haber importado varios de ellos al navegador, y entonces, si que se tendría la posibilidad de escoger con cual se quiere realizar la identificación. Tras haber seleccionado el certificado que se quiere utilizar, se ha accedido sin problema al sitio web.

---

<sup>1</sup>Public Key Infrastructure

<sup>2</sup>Online Certificate Status Protocol

<sup>3</sup>Certificate Revocation List

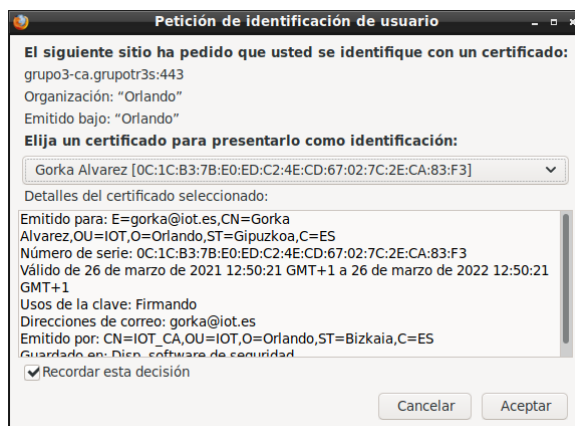


Figura 1: Petición de certificado de acceso al sitio web

Con esto solo se ha podido visualizar la correcta implementación previa de la configuración de apache y nuestra PKI. Sin embargo, se ha procedido a repetir la misma acción pero esta vez capturando el tráfico con la aplicación **Wireshark**.

En la captura deberían aparecer los intentos de conexión al OCSF por parte del navegador

1.0.00000000	127.0.0.1	127.0.0.1	TCP	74 50260 → 443 [SYN] Seq=0 Win=4096 Len=0 MSS=65495 SACK_PERM=1 TSval=54139059 TSecr=0 WS=128
2.0.000018462	127.0.0.1	127.0.0.1	TCP	74 443 → 50260 [SYN, ACK] Seq=0 Ack=1 Win=4096 Len=0 MSS=65495 SACK_PERM=1 TSval=2150838133 TSecr=54139059 WS=128
3.0.000032184	127.0.0.1	127.0.0.1	TCP	66 50260 → 443 [ACK] Seq=1 Ack=1 Win=43776 Len=0 TSval=54139059 TSecr=2150838133
4.0.001739100	127.0.0.1	127.0.0.1	TLSv1.2	583 Client Hello
5.0.001752624	127.0.0.1	127.0.0.1	TCP	66 443 → 50260 [ACK] Seq=1 Ack=518 Win=44800 Len=0 TSval=2150838135 TSecr=54139061
6.0.003767513	127.0.0.1	127.0.0.1	TLSv1.2	2717 Server Hello, Certificate, Server Key Exchange, Certificate Request, Server Hello Done
7.0.003776697	127.0.0.1	127.0.0.1	TCP	66 50260 → 443 [ACK] Seq=518 Ack=2652 Win=174720 Len=0 TSval=54139063 TSecr=2150838137
8.1.916587050	127.0.0.1	127.0.0.1	TCP	66 50260 → 443 [FIN, ACK] Seq=518 Ack=2652 Win=174720 Len=0 TSval=54140976 TSecr=2150838137
9.1.916795335	127.0.0.1	127.0.0.1	TCP	66 443 → 50260 [FIN, ACK] Seq=2652 Ack=519 Win=44800 Len=0 TSval=2150840050 TSecr=54140976
10.1.916809711	127.0.0.1	127.0.0.1	TCP	66 50260 → 443 [ACK] Seq=519 Ack=2653 Win=174720 Len=0 TSval=54140976 TSecr=2150840050

Figura 2: Captura Wireshark

Al analizar la captura se ha podido observar mejor que es lo que estaba pasando. En la comunicación únicamente toman parte dos “usuarios”, el navegador del cliente y el servidor web.

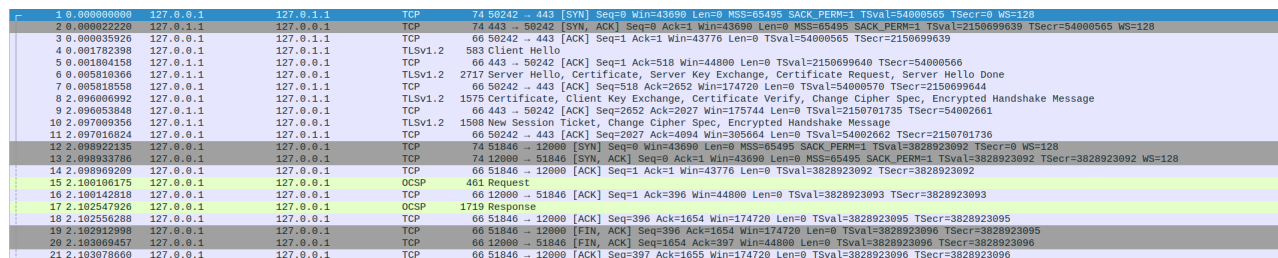
Primeramente se establece la conexión TCP y después viene el **SSL handshake**. El navegador empieza la comunicación enviando un **Client Hello**, el cual contiene la versión de SSL que puede utilizar y los algoritmos criptográficos que pueden ser utilizado para la comunicación. El servidor web le responde con un **Server Hello**, el cual contiene la versión final de **SSL** que se va a utilizar para establecer la conexión y el algoritmo criptográfico a utilizar. En este caso se ha utilizado **TLS v1.2** y el cifrado **Cipher Suite: TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256**. A su vez el servidor también envía su certificado, ~~el estado del certificado~~, intercambio de claves del servidor y por último un **Server Hello Done**.

El estado del certificado, solo puede aparecer si estamos con OCSF Stapling, no con regular OCSF

El estado del certificado indica la validez del certificado, es decir, si este está revocado o no, para ello deberá de contactar con el servidor **OCSF**, aunque en este caso no será posible ya que no se tiene el servidor en marcha. Al no haber podido contactar con el servidor **OCSF**, debería de haber ocurrido un error, no obstante, ha dado el certificado por válido y ha permitido el acceso a la página. Esto se debe a que antes los navegadores mostraban un mensaje de error, indicando que no podían contactar con el **OCSF** 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. Un ataque podría bloquear la conexión al **OCSF** responder y utilizar un certificado revocado. Es por ello, que para verificar certificados se propone el uso de **OCSF Stapling** y **Must-Staple**.

El siguiente punto de este laboratorio se basa en poner el marcha el servidor **OCSF** y comprobar con **Wireshark** cuáles son los mensajes intercambiados. Para ello se ha procedido a abrir una nueva terminal en nuestra máquina virtual y ejecutar le siguiente comando para poner el marcha nuestro servidor **OCSF** `openssl ocsf -port 12000 -index db/root-ca.index.txt -CA crt/root-ca.crt.pem -rsigner crt/ocsp-server.crt.pem -rkey private/ocsp-server.key -text -out log.txt`.

Seguidamente se ha puesto a trabajar *Wireshark* para capturar la trama y poder ver que ha pasado cuando nos conectamos a nuestra página con nuestro certificado.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	127.0.0.1	127.0.0.1	TCP	74	58242 → 443 [SYN] Seq=0 Win=43696 Len=0 MSS=65495 SACK_PERM=1 TSval=54000565 TSecr=0 WS=128
2	0.000022220	127.0.0.1	127.0.0.1	TCP	74	443 → 58242 [SYN, ACK] Seq=0 Ack=1 Win=43696 Len=0 MSS=65495 SACK_PERM=1 TSval=2150699639 TSecr=54000565 WS=128
3	0.000035920	127.0.0.1	127.0.0.1	TCP	66	58242 → 443 [ACK] Seq=1 Ack=1 Win=43776 Len=0 TSval=54000565 TSecr=2150699639
4	0.001782398	127.0.0.1	127.0.0.1	TLShv1.2	583	Client Hello
5	0.001804158	127.0.0.1	127.0.0.1	TCP	66	443 → 58242 [ACK] Seq=1 Ack=518 Win=44800 Len=0 TSval=2150699640 TSecr=54000565
6	0.005810366	127.0.0.1	127.0.0.1	TLShv1.2	2717	Server Hello, Certificate, Server Key Exchange, Certificate Request, Server Hello Done
7	0.005810558	127.0.0.1	127.0.0.1	TCP	66	58242 → 443 [ACK] Seq=518 Ack=2052 Win=174720 Len=0 TSval=54000570 TSecr=2150699644
8	0.006000992	127.0.0.1	127.0.0.1	TLShv1.2	1575	Certificate, Client Key Exchange, Certificate Verify, Change Cipher Spec, Encrypted Handshake Message
9	0.006053848	127.0.0.1	127.0.0.1	TCP	66	443 → 58242 [ACK] Seq=2052 Ack=2027 Win=175744 Len=0 TSval=2150701735 TSecr=54000565
10	0.007009356	127.0.0.1	127.0.0.1	TLShv1.2	1508	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
11	0.007010824	127.0.0.1	127.0.0.1	TCP	66	58242 → 443 [ACK] Seq=2027 Ack=4094 Win=305664 Len=0 TSval=54002062 TSecr=2150701736
12	0.008922136	127.0.0.1	127.0.0.1	TCP	74	51846 → 12080 [SYN] Seq=0 Win=43696 Len=0 MSS=65495 SACK_PERM=1 TSval=3828923092 TSecr=0 WS=128
13	0.008933786	127.0.0.1	127.0.0.1	TCP	74	12080 → 51846 [SYN, ACK] Seq=0 Ack=1 Win=43696 Len=0 MSS=65495 SACK_PERM=1 TSval=3828923092 TSecr=3828923092 WS=128
14	0.008969209	127.0.0.1	127.0.0.1	TCP	66	51846 → 12080 [ACK] Seq=1 Ack=1 Win=43776 Len=0 TSval=3828923092 TSecr=3828923092
15	0.100106175	127.0.0.1	127.0.0.1	OCSP	461	Request
16	0.100142818	127.0.0.1	127.0.0.1	TCP	66	12080 → 51846 [ACK] Seq=1 Ack=396 Win=44800 Len=0 TSval=3828923093 TSecr=3828923093
17	0.102547926	127.0.0.1	127.0.0.1	OCSP	1719	Response
18	0.102556288	127.0.0.1	127.0.0.1	TCP	66	51846 → 12080 [ACK] Seq=396 Ack=1654 Win=174720 Len=0 TSval=3828923095 TSecr=3828923095
19	0.102912998	127.0.0.1	127.0.0.1	TCP	66	51846 → 12080 [FIN, ACK] Seq=1654 Ack=1654 Win=174720 Len=0 TSval=3828923096 TSecr=3828923095
20	0.103000457	127.0.0.1	127.0.0.1	TCP	66	12080 → 51846 [FIN, ACK] Seq=1654 Ack=397 Win=44800 Len=0 TSval=3828923096 TSecr=3828923096
21	0.103078669	127.0.0.1	127.0.0.1	TCP	66	51846 → 12080 [ACK] Seq=397 Ack=1655 Win=174720 Len=0 TSval=3828923096 TSecr=3828923096

Figura 3: Captura *Wireshark*

Al analizar la trama se ha podido observar que después del *handshake* el cliente, en este caso el navegador que se ha utilizado, utiliza un puerto distinto para ponerse en contacto con nuestro servidor *OCSP*. Por lo que se ha llegado a la conclusión de que aplicando esta metodología se está dejando que el propio cliente se encarga de verificar el certificado del servidor, en este caso del servidor web Apache. Para ponerse en contacto envía un *Request* al servidor *OCSP* y éste le responde con un *Response*. En este caso el certificado que se ha utilizado no estaba revocado por lo que la respuesta del *OCSP* ha sido satisfactoria.

Para concluir con esta primera parte se pide realizar el mismo análisis pero esta vez utilizando un certificado de servidor revocado. Por tanto, el primer paso a realizar es revocar el certificado. Para ello hemos utilizado el comando `openssl ca -revoke crt/server.crt.pem -crl_reason superseded -config conf_file.cnf`. Una vez hecho eso se ha vuelto a acceder a nuestra página web, aunque esta vez sin éxito. Ahora vamos a mirar la captura a ver que ha sucedido.

Al igual que antes, es el navegador el que se ha puesto en contacto con el servidor *OCSP*. De nuevo le envía un *request* con el certificado del servidor y éste le devuelve un *response*, pero esta vez el *responseStatus* es fallo. Por lo que se puede concluir que el servidor *OCSP* ha realizado correctamente su trabajo.

## 2.2. OCSP Stapling

El servidor *OCSP Stapling* soluciona algunos de los problemas del servidor *OCSP* convencional, ya que obliga al servidor a emitir las consulta *OCSP* periódicamente. En lugar de hacer que el usuario realice la verificación de validez, el servidor web le presentará la respuesta *OCSP* durante el protocolo *TLS*. Dado que el servidor almacena en cachés la respuesta, las CA ya no estará inundada de solicitudes *OCSP* y, como usuario, no es necesario que se comunique con un tercero para verificar un certificado que beneficie su privacidad.

El primer paso a dar es activar el módulo *mod\_sosache\_shmcb* ejecutando el comando `a2enmod sosache_shmcb`. Después se han añadido las líneas correspondientes al fichero del sitio virtual que permiten hacer funcionar el servidor *OCSP Stapling*. Una vez añadidas, se ha puesto en marcha el servidor ejecutando el comando antes visto `opensslslosp -port 12000 -index db/root-ca.index.txt -CA crt/root-ca.crt.pem -rsigner crt/ocsp-server.crt.pem -rkey private/ocsp-server.key -text -out log.txt`. A continuación se ha puesto a capturar de nuevo el *Wireshark*.

1	0.000000000	127.0.0.1	127.0.0.1	TCP	74 50340 → 443 [SYN] Seq=0 Win=43690 Len=0 MSS=65495 SACK_PERM=1 TSval=
2	0.000001202	127.0.0.1	127.0.0.1	TCP	74 443 → 50340 [SYN, ACK] Seq=0 Ack=1 Win=43690 Len=0 MSS=65495 SACK_P
3	0.000033154	127.0.0.1	127.0.0.1	TCP	66 50340 → 443 [ACK] Seq=1 Ack=1 Win=43776 Len=0 TSval=54734071 TSsecr=
4	0.001740948	127.0.0.1	127.0.0.1	TLSv1.2	583 Client Hello
5	0.001764048	127.0.0.1	127.0.0.1	TCP	66 443 → 50340 [ACK] Seq=1 Ack=518 Win=44800 Len=0 TSval=2151433147 TS
6	0.002936351	:::1	:::1	TCP	94 56356 → 12000 [SYN] Seq=0 Win=43690 Len=0 MSS=65476 SACK_PERM=1 TSv
7	0.002946420	:::1	:::1	TCP	74 12000 → 56356 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
8	0.002992730	127.0.0.1	127.0.0.1	TCP	74 51946 → 12000 [SYN] Seq=0 Win=43690 Len=0 MSS=65495 SACK_PERM=1 TSV
9	0.003095686	127.0.0.1	127.0.0.1	TCP	74 12000 → 51946 [SYN, ACK] Seq=0 Ack=1 Win=43690 Len=0 MSS=65495 SACK
10	0.003015070	127.0.0.1	127.0.0.1	TCP	66 51946 → 12000 [ACK] Seq=1 Ack=1 Win=43776 Len=0 TSval=3829654502 TS
11	0.003078893	127.0.0.1	127.0.0.1	OCSP	251 Request
12	0.003094521	127.0.0.1	127.0.0.1	TCP	66 12000 → 51946 [ACK] Seq=1 Ack=186 Win=44800 Len=0 TSval=3829654502
13	0.005599331	127.0.0.1	127.0.0.1	OCSP	1719 Response
14	0.005607913	127.0.0.1	127.0.0.1	TCP	66 51946 → 12000 [ACK] Seq=186 Ack=1654 Win=174720 Len=0 TSval=3829654
15	0.006113848	127.0.0.1	127.0.0.1	TCP	66 12000 → 51946 [FIN, ACK] Seq=1654 Ack=186 Win=44800 Len=0 TSval=382
16	0.006242315	127.0.0.1	127.0.0.1	TCP	66 51946 → 12000 [FIN, ACK] Seq=186 Ack=1655 Win=174720 Len=0 TSval=38
17	0.006253889	127.0.0.1	127.0.0.1	TCP	66 12000 → 51946 [ACK] Seq=1655 Ack=187 Win=44800 Len=0 TSval=38296545
18	0.008940667	127.0.0.1	127.0.0.1	TLSv1.2	4162 Server Hello, Certificate, Certificate Status, Server Key Exchange
19	0.008978582	127.0.0.1	127.0.0.1	TCP	66 50340 → 443 [ACK] Seq=518 Ack=4097 Win=174720 Len=0 TSval=54734080
20	0.009013203	127.0.0.1	127.0.0.1	TLSv1.2	209 Certificate Request, Server Hello Done
21	0.009017076	127.0.0.1	127.0.0.1	TCP	66 50340 → 443 [ACK] Seq=518 Ack=4240 Win=182912 Len=0 TSval=54734080
22	1.001956276	127.0.0.1	127.0.0.1	TLSv1.2	1575 Certificate, Client Key Exchange, Certificate Verify, Change Cipher
23	1.001177080	127.0.0.1	127.0.0.1	TCP	66 443 → 50340 [ACK] Seq=4240 Ack=2027 Win=175744 Len=0 TSval=21514348
24	1.002444075	127.0.0.1	127.0.0.1	TLSv1.2	1508 New Session Ticket, Change Cipher Spec, Encrypted Handshake Message

Figura 4: Captura Wireshark

Ahora si aparece Certificate Status

Al analizar la captura se ha podido observar cual era el funcionamiento. Respecto al anterior han cambiado el orden los pasos. Esta vez es el servidor web es que se pone en contacto con el servidor *OCSP*. También cabe destacar que se pone en contacta con el servidor antes del *Server Hello* del servidor. En nuestro caso como el certificado no estaba revocado el *OCSP response* ha sido correcto y hemos podido acceder a la página. Después de la respuesta se completa el resto del *handshake*.

A continuación, comprobaremos lo que ocurre al revocar el certificado del sitio web, en nuestro caso, el certificado denominado *server.crt*. En las figuras 6 y 5, se puede ver el certificado del servidor que se ha revocado, su número de serie y la razón por la que se ha revocado el certificado.

srdsi-ssl.conf	conf_file.cnf	root-ca.index.txt	ocsp-log.txt
V	220301190412Z	0C1CB37BE0EDC24ECD67027C2ECA83F0	unknown /C=ES/ST=Bizkaia/O=Orlando/OU=IOT/CN=IOT_CA
V	220301190603Z	0C1CB37BE0EDC24ECD67027C2ECA83F1	unknown /C=ES/ST=Bizkaia/O=Orlando/OU=IOT/CN=IOT_SubCA
R	220326114839Z	210326125026Z, superseded	0C1CB37BE0EDC24ECD67027C2ECA83F2
V	220326115021Z	0C1CB37BE0EDC24ECD67027C2ECA83F3	unknown /C=ES/ST=Gipuzkoa/O=Orlando/OU=IOT/CN=Gorka Al
V	220326115058Z	0C1CB37BE0EDC24ECD67027C2ECA83F4	unknown /C=ES/ST=Gipuzkoa/O=Orlando/OU=IOT/CN=www.iot-
V	220326144309Z	0C1CB37BE0EDC24ECD67027C2ECA83F5	unknown /C=ES/ST=Bizkaia/O=Orlando/OU=IOT/CN=grupo3-ca
V	220326145933Z	0C1CB37BE0EDC24ECD67027C2ECA83F6	unknown /C=ES/ST=Bizkaia/O=Orlando/OU=IOT/CN=grupo3-ca

Figura 5: Base de Datos de la CA raíz

server\_revoked.crt

server@grupotr3s.com

Nombre del emisor

C (País): ES

ST (Estado): Bizkaia

O (Organización): Orlando

OU (Unidad de organización): IOT

CN (Nombre común): IOT\_CA

Certificado emitido

Versión: 3

Número de serie: 0C 1C B3 7B E0 ED C2 4E CD 67 02 7C 2E CA 83 F2

No es válido antes de: 2021-03-26

No es válido después de: 2022-03-26

Huellas de certificados

SHA1: 40 87 53 FD C7 A0 3B 90 A7 07 72 CE 06 12

Cerrar

Importar

Figura 6: Número de serie del certificado del servidor

A continuación, se reiniciarán el servicio apache de nuestro sitio web y el servidor *OCSP*, de manera que se actualice la base de datos de trabajo del servidor. Una vez hecho esto, se procederá a conectarse al sitio web. Sin embargo, y tal como se ve en la figura 7, no se permitirá acceso al sitio web debido a que la CA raíz ha revocado el certificado de este sitio.



Figura 7: Error de acceso por revocación del certificado del sitio web

### 2.3. Must-Staple

Este tipo de servidor OCSP es un simple añadido al método *OCSP Stapling* anterior. Si usamos este método, debermos agregar una extensión en el certificado del servidor que obligará el uso del *OCSP Stapling*, de esta manera, solo se permitirán conexiones al sitio web en caso de que todo funcione correctamente (CA, cliente y sitio web).

La primera de las configuraciones a realizar para activar este método es configurar la extensión aplicada al certificado del servidor web seguro. Para ello, se modificará la extensión relativa al servidor, *server\_ext*, en el fichero de configuración de la PKI, *conf.file.cnf*.

```
[server_ext]
authorityInfoAccess      = @issuer_info
authorityKeyIdentifier    = keyid:always
basicConstraints          = critical,CA:false
crlDistributionPoints     = @crl_info
extendedKeyUsage          = clientAuth,serverAuth
keyUsage                  = critical,digitalSignature,keyEncipherment
subjectKeyIdentifier      = hash
tlsfeature                 = status_request
```

Figura 8: Nueva directiva en la extensión del servidor

A continuación, utilizando la solicitud, y clave privada previamente creadas, se generará un nuevo certificado para el sitio web, `openssl ca -in certreqs/server.csr -out crt/server.crt -config conf.file.cnf -extensions server_ext`. Cuando aparezca el mensaje de confirmación de firma, se comprobará que efectivamente aparezca esta nueva directriz.



```
X509v3 Extended Key Usage:
    TLS Web Client Authentication, TLS Web Server Authentication
X509v3 Key Usage: critical
    Digital Signature, Key Encipherment
X509v3 Subject Key Identifier:
    C7:45:F6:2C:23:C9:0E:62:AF:B9:05:BA:8E:D9:60:FF:41:83:77:18
    TLS Feature:
    status_request
Certificate is to be certified until Mar 26 14:59:33 2022 GMT (365 days)
Sign the certificate? [y/n]:
```

Figura 9: Comprobación de extensiones durante la firma del certificado del sitio web

Puesto que el nombre con el que se ha guardado el nuevo certificado del sitio web es el mismo que en los apartados anteriores, únicamente se tendrá que mover este certificado a la dirección `/var/www/srdsi-lab/certificados/`, y reiniciar el servidor apache para aplicar los nuevos cambios.

El siguiente paso es comprobar que efectivamente funciona la nueva característica que se ha agregado al sitio web. Para ello, se tratará de acceder de dos formas diferentes, la primera, será con el servidor OCSP desactivado. El resultado esperado es un error de conexión debido a que el sitio web no puede conectarse con el servidor OCSP, por lo que rechazará cualquier intento de conexión.



Figura 10: Error de acceso debido a la indisponibilidad del servidor OCSP

Tal como se ve en la figura 10, no se ha sido capaz de acceder al sitio web. El navegador muestra un mensaje con código de error, `SEC_ERROR_OCSP_TRY_SERVER_LATER`, el cual indica que el servidor OCSP no se encuentra actualmente disponible, por lo que no se podrán validar los certificados, y por consecuencia, acceder al sitio web.

Tras activar el servidor OCSP de nuevo, se tratará de acceder al sitio web de nuevo. Sin embargo, antes de hacer el acceso, y después de activar el servidor OCSP, se reinició el servidor apache para asegurar el correcto funcionamiento. En la figura 11, se puede observar como se ha podido acceder correctamente tras la activación del servidor OCSP.





Figura 11: Acceso correcto tras activar el OCSP