

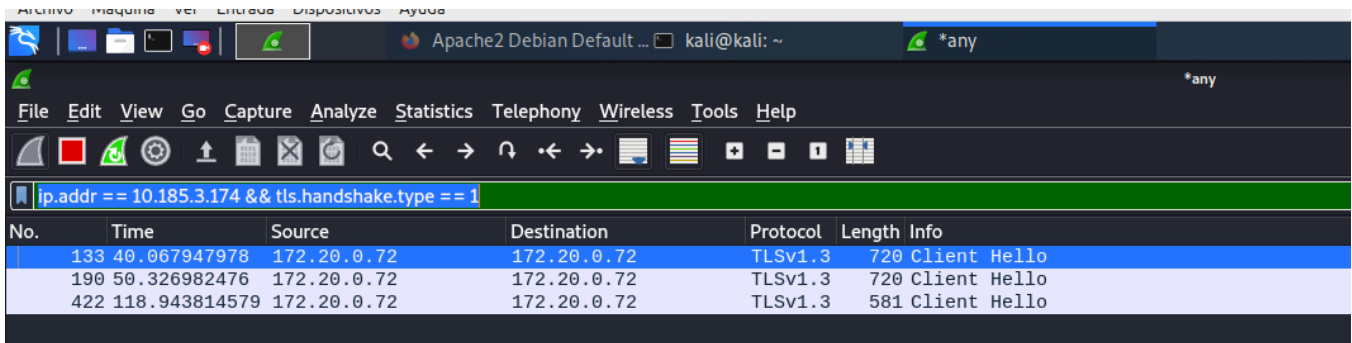
Todos los integrantes del Grupo 06

Lab de TLS (Transport layer security)

Abrimos el wireshark (modo any). Filtramos en wireshark usando la ip.addr (igual a la misma que configuramos el servidor apache) y obtenemos los paquetes client Hello

En mi caso el filtro es:

ip.addr == 10.185.3.174 && tls.handshake.type == 1

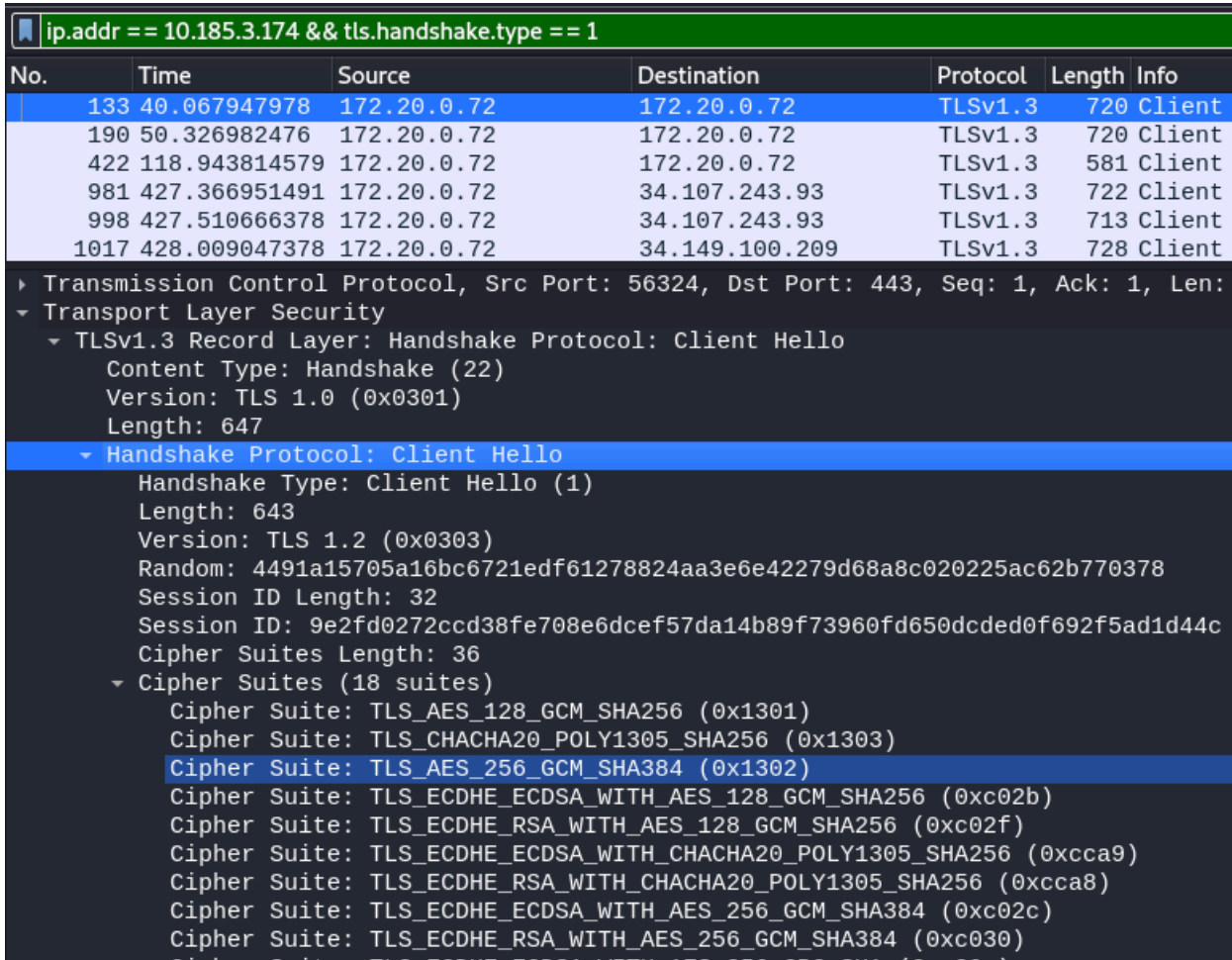


The image shows the Wireshark interface with the filter `ip.addr == 10.185.3.174 && tls.handshake.type == 1` applied. The packet list shows three packets of type TLSv1.3 Client Hello.

No.	Time	Source	Destination	Protocol	Length	Info
133	40.067947978	172.20.0.72	172.20.0.72	TLSv1.3	720	Client Hello
190	50.326982476	172.20.0.72	172.20.0.72	TLSv1.3	720	Client Hello
422	118.943814579	172.20.0.72	172.20.0.72	TLSv1.3	581	Client Hello

1. Ver el paquete de Client Hello. ¿qué cifradores ofrece el cliente? ¿que contiene la extensión de SSL server name? ¿para que sirve?

Dentro del paquete client hello uno de los campos es “Cipher Suites” (Conjunto de cifrados) este campo muestra el conjunto de algoritmo de encriptación que el cliente ofrece al servidor para negociar una conexión segura. Aquí se muestran algunos algoritmos de encriptación:



The image shows the details pane of a TLSv1.3 Client Hello packet. The 'Cipher Suites' field is expanded, showing a list of supported cipher suites.

No.	Time	Source	Destination	Protocol	Length	Info
133	40.067947978	172.20.0.72	172.20.0.72	TLSv1.3	720	Client
190	50.326982476	172.20.0.72	172.20.0.72	TLSv1.3	720	Client
422	118.943814579	172.20.0.72	172.20.0.72	TLSv1.3	581	Client
981	427.366951491	172.20.0.72	34.107.243.93	TLSv1.3	722	Client
998	427.510666378	172.20.0.72	34.107.243.93	TLSv1.3	713	Client
1017	428.009047378	172.20.0.72	34.149.100.209	TLSv1.3	728	Client

Transmission Control Protocol, Src Port: 56324, Dst Port: 443, Seq: 1, Ack: 1, Len: 720

Transport Layer Security

- TLSv1.3 Record Layer: Handshake Protocol: Client Hello
 - Content Type: Handshake (22)
 - Version: TLS 1.0 (0x0301)
 - Length: 647
- Handshake Protocol: Client Hello
 - Handshake Type: Client Hello (1)
 - Length: 643
 - Version: TLS 1.2 (0x0303)
 - Random: 4491a15705a16bc6721edf61278824aa3e6e42279d68a8c020225ac62b770378
 - Session ID Length: 32
 - Session ID: 9e2fd0272ccd38fe708e6dcef57da14b89f73960fd650dcded0f692f5ad1d44c
 - Cipher Suites Length: 36
 - Cipher Suites (18 suites)
 - Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)
 - Cipher Suite: TLS_CHACHA20_POLY1305_SHA256 (0x1303)
 - Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
 - Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)
 - Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
 - Cipher Suite: TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca9)
 - Cipher Suite: TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xc030)
 - Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc031)
 - Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc032)
 - Cipher Suite: TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8)
 - Cipher Suite: TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xc033)
 - Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CCM_SHA256 (0xc034)
 - Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CCM_SHA256 (0xc035)
 - Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CCM_SHA384 (0xc036)
 - Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CCM_SHA384 (0xc037)
 - Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc038)
 - Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc039)
 - Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc03a)
 - Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc03b)

La extensión de “server_name” contiene lo que se ve en la imagen:

```
Extensions Length: 534
  ▾ Extension: server_name (len=23)
    Type: server_name (0)
    Length: 23
    ▾ Server Name Indication extension
      Server Name list length: 21
      Server Name Type: host_name (0)
      Server Name length: 18
      Server Name: apache.empresa.com
  ▸ Extension: extended_master_secret (len=0)
  ▸ Extension: renegotiation_info (len=1)
```

La extensión SNI (Server name indication) sirve para que el cliente indique a que hostName esta tratando de conectarse al inicio del proceso de handshake.

Permite que un dispositivo de usuario abra una conexión segura con (en este caso) **https://apache.empresa.com/** incluso si ese sitio web está alojado en el mismo lugar (misma dirección IP) que <https://www.something.com>, <https://www.another-website.com>, otras paginas básicamente.

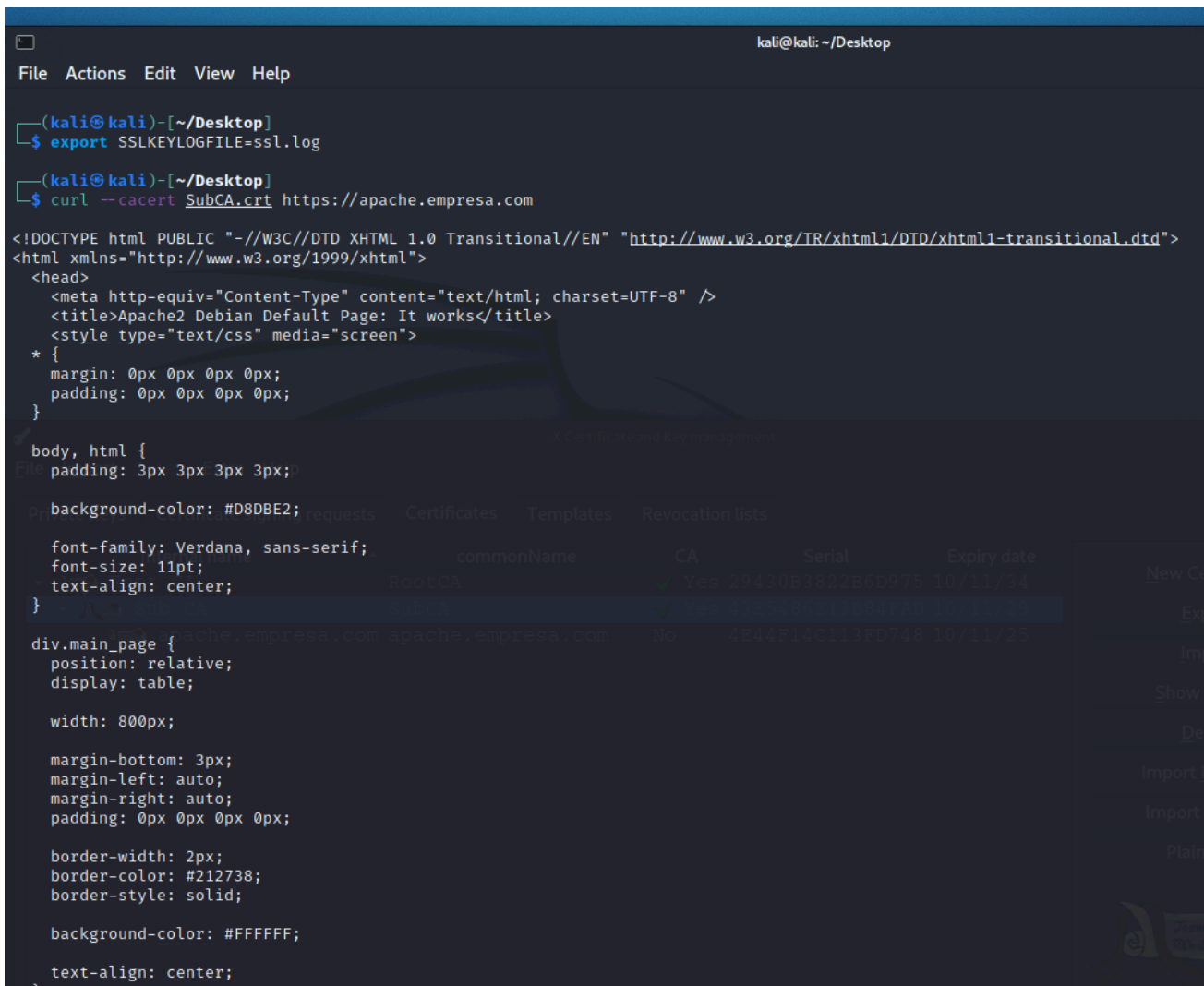
2. Ver el paquete de ServerHello. Cual es el Cifrador que selecciono el Servidor?

ip.addr == 172.20.0.72 && tls.handshake.type == 2

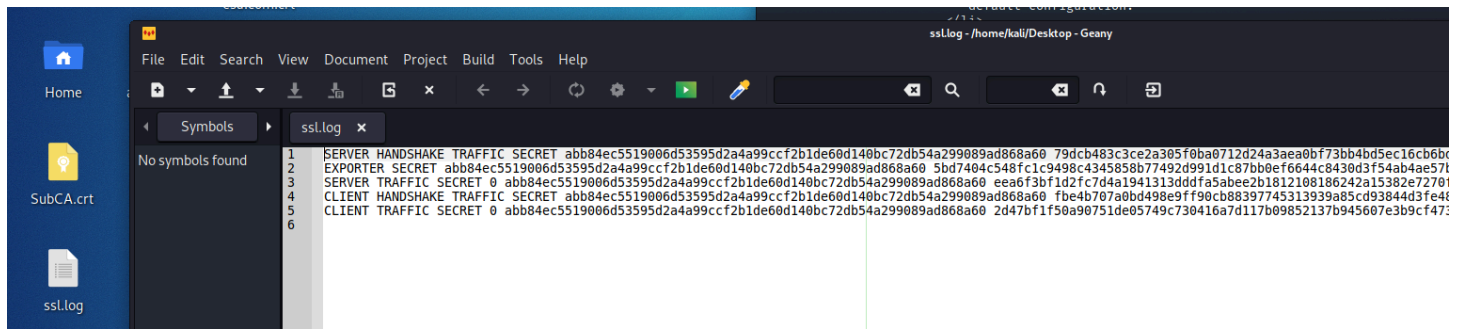
No.	Time	Source	Destination	Protocol	Length	Info
10	1.978812575	172.20.0.72	172.20.0.72	TLSv1.3	312	Server Hello, Ch
5295	323.482833435	142.251.134.10	172.20.0.72	TLSv1.3	836	Server Hello, Ch

▸ Frame 10: 312 bytes on wire (2496 bits), 312 bytes captured (2496 bits) on interface any, id
▸ Linux cooked capture v1
▸ Internet Protocol Version 4, Src: 172.20.0.72, Dst: 172.20.0.72
▸ Transmission Control Protocol, Src Port: 443, Dst Port: 56342, Seq: 1, Ack: 653, Len: 244
▾ Transport Layer Security
 ▾ TLSv1.3 Record Layer: Handshake Protocol: Server Hello
 Content Type: Handshake (22)
 Version: TLS 1.2 (0x0303)
 Length: 128
 ▾ Handshake Protocol: Server Hello
 Handshake Type: Server Hello (2)
 Length: 124
 Version: TLS 1.2 (0x0303)
 Random: 8a2276090432f196ea3873e827ef18af792a1fad55e0486e4012f3bb93334968
 Session ID Length: 32
 Session ID: 8c9bb1b94cefdb4d2e8cf412da6569b5f05c583e61976b4593157aad2143a51
 Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)
 Compression Method: null (0)
 Extensions Length: 52
 ▸ Extension: supported_versions (len=2)
 ▸ Extension: key_share (len=36)
 ▸ Extension: pre_shared_key (len=2)

Exportamos el .crt de la Sub_CA:



Se nos crea el archivo ssl.log y lo observamos el sgt contenido:



Trafico de wireshark:

ip.addr == 172.20.0.72

No.	Time	Source	Destination	Protocol	Length	Info
399	13.747050869	172.20.0.72	172.20.0.72	TCP	76	56356 → 443 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=979815044 TSecr=0 WS=128
400	13.747066415	172.20.0.72	172.20.0.72	TCP	76	443 → 56356 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM=1 TSval=979815044 TSecr=0
401	13.747079175	172.20.0.72	172.20.0.72	TCP	68	56356 → 443 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=979815044 TSecr=979815044
402	13.748913322	172.20.0.72	172.20.0.72	TLSv1.3	585	Client Hello
403	13.748978015	172.20.0.72	172.20.0.72	TCP	68	443 → 56356 [ACK] Seq=1 Ack=518 Win=65024 Len=0 TSval=979815046 TSecr=979815046
404	13.752825621	172.20.0.72	172.20.0.72	TLSv1.3	1976	Server Hello, Change Cipher Spec, Application Data, Application Data, Application Data, Application Data
405	13.752842426	172.20.0.72	172.20.0.72	TCP	68	56356 → 443 [ACK] Seq=518 Ack=1909 Win=64000 Len=0 TSval=979815050 TSecr=979815050
406	13.754041015	172.20.0.72	172.20.0.72	TLSv1.3	148	Change Cipher Spec, Application Data
407	13.754057162	172.20.0.72	172.20.0.72	TCP	68	443 → 56356 [ACK] Seq=1909 Ack=598 Win=65536 Len=0 TSval=979815051 TSecr=979815051
408	13.754233245	172.20.0.72	172.20.0.72	TLSv1.3	172	Application Data
409	13.754237675	172.20.0.72	172.20.0.72	TCP	68	443 → 56356 [ACK] Seq=1909 Ack=702 Win=65536 Len=0 TSval=979815051 TSecr=979815051
410	13.754338778	172.20.0.72	172.20.0.72	TLSv1.3	371	Application Data
411	13.754344348	172.20.0.72	172.20.0.72	TCP	68	56356 → 443 [ACK] Seq=702 Ack=2212 Win=65280 Len=0 TSval=979815051 TSecr=979815051
412	13.754460838	172.20.0.72	172.20.0.72	TLSv1.3	371	Application Data
413	13.754464384	172.20.0.72	172.20.0.72	TCP	68	56356 → 443 [ACK] Seq=702 Ack=2515 Win=65024 Len=0 TSval=979815052 TSecr=979815052
414	13.754762409	172.20.0.72	172.20.0.72	TLSv1.3	11068	Application Data, Application Data
415	13.754777559	172.20.0.72	172.20.0.72	TCP	68	56356 → 443 [ACK] Seq=702 Ack=13515 Win=59520 Len=0 TSval=979815052 TSecr=979815052
416	13.755293185	172.20.0.72	172.20.0.72	TLSv1.3	92	Application Data
417	13.755305820	172.20.0.72	172.20.0.72	TCP	68	443 → 56356 [ACK] Seq=13515 Ack=726 Win=65536 Len=0 TSval=979815052 TSecr=979815052
418	13.755366583	172.20.0.72	172.20.0.72	TCP	68	56356 → 443 [FIN, ACK] Seq=726 Ack=13515 Win=65536 Len=0 TSval=979815052 TSecr=979815052
419	13.755410879	172.20.0.72	172.20.0.72	TLSv1.3	92	Application Data
420	13.755427912	172.20.0.72	172.20.0.72	TCP	56	56356 → 443 [RST] Seq=727 Win=0 Len=0

Importamos el archivo ssl.log y observamos en el wireshark el tráfico descifrado:

ip.addr == 172.20.0.72

No.	Time	Source	Destination	Protocol	Length	Info
399	13.747050869	172.20.0.72	172.20.0.72	TCP	76	56356 → 443 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=979815044 TSecr=0 WS=128
400	13.747066415	172.20.0.72	172.20.0.72	TCP	76	443 → 56356 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM=1 TSval=979815044 TSecr=0
401	13.747079175	172.20.0.72	172.20.0.72	TCP	68	56356 → 443 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=979815044 TSecr=979815044
402	13.748913322	172.20.0.72	172.20.0.72	TLSv1.3	585	Client Hello
403	13.748978015	172.20.0.72	172.20.0.72	TCP	68	443 → 56356 [ACK] Seq=1 Ack=518 Win=65024 Len=0 TSval=979815046 TSecr=979815046
404	13.752825621	172.20.0.72	172.20.0.72	TLSv1.3	1976	Server Hello, Change Cipher Spec, Encrypted Extensions, Application Data, Application Data, Application Data, Application Data
405	13.752842426	172.20.0.72	172.20.0.72	TCP	68	56356 → 443 [ACK] Seq=518 Ack=1909 Win=64000 Len=0 TSval=979815050 TSecr=979815050
406	13.754041015	172.20.0.72	172.20.0.72	TLSv1.3	148	Change Cipher Spec, Finished
407	13.754057162	172.20.0.72	172.20.0.72	TCP	68	443 → 56356 [ACK] Seq=1909 Ack=598 Win=65536 Len=0 TSval=979815051 TSecr=979815051
408	13.754233245	172.20.0.72	172.20.0.72	HTTP	172	GET / HTTP/1.1
409	13.754237675	172.20.0.72	172.20.0.72	TCP	68	443 → 56356 [ACK] Seq=1909 Ack=702 Win=65536 Len=0 TSval=979815051 TSecr=979815051
410	13.754338778	172.20.0.72	172.20.0.72	TLSv1.3	371	New Session Ticket
411	13.754344348	172.20.0.72	172.20.0.72	TCP	68	56356 → 443 [ACK] Seq=702 Ack=2212 Win=65280 Len=0 TSval=979815051 TSecr=979815051
412	13.754460838	172.20.0.72	172.20.0.72	TLSv1.3	371	New Session Ticket
413	13.754464384	172.20.0.72	172.20.0.72	TCP	68	56356 → 443 [ACK] Seq=702 Ack=2515 Win=65024 Len=0 TSval=979815052 TSecr=979815052
414	13.754762409	172.20.0.72	172.20.0.72	HTTP	11068	HTTP/1.1 200 OK (text/html)
415	13.754777559	172.20.0.72	172.20.0.72	TCP	68	56356 → 443 [ACK] Seq=702 Ack=13515 Win=59520 Len=0 TSval=979815052 TSecr=979815052
416	13.755293185	172.20.0.72	172.20.0.72	TLSv1.3	92	Alert (Level: Warning, Description: Close Notice)
417	13.755305820	172.20.0.72	172.20.0.72	TCP	68	443 → 56356 [ACK] Seq=13515 Ack=726 Win=65536 Len=0 TSval=979815052 TSecr=979815052
418	13.755366583	172.20.0.72	172.20.0.72	TCP	68	56356 → 443 [FIN, ACK] Seq=726 Ack=13515 Win=0 Len=0 TSval=979815052 TSecr=979815052
419	13.755410879	172.20.0.72	172.20.0.72	TLSv1.3	92	Alert (Level: Warning, Description: Close Notice)
420	13.755427912	172.20.0.72	172.20.0.72	TCP	56	56356 → 443 [RST] Seq=727 Win=0 Len=0

Configurar Autenticación de 2 Vías

Creamos los archivos index.php y seguro.php en la ruta indicada y copiamos el contenido indicado:

```
Apache2 Debian Default ... kali@kali: ~/Desktop *any

File Actions Edit View Help

(kali@kali)-[~/Desktop]
$ sudo nano /var/www/html/index.php
[sudo] password for kali:

(kali@kali)-[~/Desktop]
$ cat /var/www/html/index.php
<?php
echo "SERVER=". $_SERVER['SERVER_NAME'];
echo "<br>";
echo "USUARIO DEL CERTIFICADO=".$_SERVER['SSL_CLIENT_S_DN_CN'];
?>

(kali@kali)-[~/Desktop]
$ sudo nano /var/www/html/seguro.php

(kali@kali)-[~/Desktop]
$ cat /var/www/html/seguro.php
<?php
echo "<h1>zona segura </h1>";
echo "SERVER=". $_SERVER['SERVER_NAME']. "<br>";
echo "USUARIO DEL CERTIFICADO=" .
$_SERVER['SSL_CLIENT_S_DN_CN']. "<br>";
echo "Organizacion=".$_SERVER['SSL_CLIENT_S_DN_O']. "<br>";
echo "<h2> variables</h2>";
foreach ($_SERVER as $k=>$v)
echo $k . "=>" . $v . "<br>";
?>

(kali@kali)-[~/Desktop]
$
```

Modificamos el contenido del archivo de configuración de apache:
agregando las lineas:

DirectoryIndex index.php index.html

ErrorLog \${APACHE_LOG_DIR}/SSLerror.log

CustomLog \${APACHE_LOG_DIR}/SSLaccess.log combined

```
VirtualHost *:443>
  ServerName apache.empresa.com
  SSLEngine on
  SSLCertificateFile "/home/kali/Desktop/apache.empresa.com.crt"
  SSLCertificateKeyFile "/home/kali/Desktop/apache.key"
  DirectoryIndex index.php index.html
  ErrorLog ${APACHE_LOG_DIR}/SSLerror.log
  CustomLog ${APACHE_LOG_DIR}/SSLaccess.log combined
VirtualHost>

vim: syntax=apache ts=4 sw=4 sts=4 sr noet
REQUESTS=>1
[ Read 42 lines ]
Help: ^O Write Out ^W Where Is In ^K Cut ^T Execute ^C Location
Exit ^R Read File ^N Replace ^U Paste ^J Justify ^_ Go To Line
```

Ademas hacemos un:

sudo chmod 644 /var/www/html/index.php

sudo chmod 644 /var/www/html/seguro.php

Vemos que al navegar a <https://apache.empresa.com> ahora obtenemos una nueva pagina (izquierda) y si entramos a seguro.php (derecha) nos muestra el sgt contenido:

apache.empresa.com/ x +
← → ↻ 🏠 🔒 https://apache.empresa.com 130% ... ☆ 📄 📄
SERVER=apache.empresa.com
USUARIO DEL CERTIFICADO=

apache.empresa.com/seguro.php x +
← → ↻ 🏠 🔒 https://apache.empresa.com/seguro.php 130% ... ☆ 📄 📄
zona segura
SERVER=apache.empresa.com
USUARIO DEL CERTIFICADO=
Organizacion=
variables
UNIQUE_ID=>Zy@CoE7dlMltnlEuUVnSswAAAAI
HTTPS=>on
SSL_TLS_SNI=>apache.empresa.com
HTTP_HOST=>apache.empresa.com
HTTP_USER_AGENT=>Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/
Firefox/78.0
HTTP_ACCEPT=>text/html,application/xhtml+xml,application/xml;q=0
/webp;*/*;q=0.8
HTTP_ACCEPT_LANGUAGE=>en-US,en;q=0.5
HTTP_ACCEPT_ENCODING=>gzip, deflate, br
HTTP_CONNECTION=>keep-alive
HTTP_UPGRADE_INSECURE_REQUESTS=>1
PATH=>/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
SERVER_SIGNATURE=>
Apache/2.4.46 (Debian) Server at apache.empresa.com Port 443

SERVER_SOFTWARE=>Apache/2.4.46 (Debian)
SERVER_NAME=>apache.empresa.com
SERVER_ADDR=>172.25.118.52

Autenticación de dos Vías

Creamos al Usuario1 con los sgts datos, además generamos una new key usando **RSA-2048bits**:

X Certificate and Key management

Create x509 Certificate

Source Subject Extensions Key usage Netscape Advanced Comment

Internal Name

Distinguished name

countryName	AR	organizationalUnitName	seguridad
stateOrProvinceName	CABA	commonName	Usuario1
localityName	CABA	emailAddress	usuario1@gmail.com
organizationName	seguridad		

Type	Content	Add	Delete

Private key

Usuario1 (RSA:2048 bit) ☐ Used keys too

OK Cancel

Signing

☐ Create a self signed certificate
☒ Use this Certificate for signing

Sub CA

Signature algorithm: SHA 256

Template for the new certificate: [default] TLS_client

Además le ponemos como template TLS_client (y le ponemos sub CA para firmar) .

Ahora creamos al usuario2 con los sgt datos:

X Certificate and Key management

Create x509 Certificate

Source Subject Extensions Key usage Netscape Advanced Comment

Internal Name

Distinguished name

countryName	AR	organizationalUnitName	contable_org
stateOrProvinceName	CABA	commonName	Usuario2
localityName	CABA	emailAddress	usuario2@gmail.com
organizationName	contable_org		

Type	Content

Private key

Usuario2 (RSA:2048 bit)
 ☐ Used keys too

Idem al usuario 1 le ponemos como template TLS_client usando sub_CA. Finalmente obtenemos:

X Certificate and Key management					
File Import Token Extra Help					
Private Keys Certificate signing requests Certificates Templates Revocation lists					
Internal name	commonName	CA	Serial	Expiry date	
Root CA	RootCA	✓ Yes	29430B3822B6D975	10/11/34	
Sub CA	SubCA	✓ Yes	43E5486E13B84FAD	10/11/29	
apache.empresa.com	apache.empresa.com	No	4E44F14C113FD748	10/11/25	
Usuario2	Usuario2		19BE42909678FCE2	11/9/25	
Usuario1	Usuario1		2C5145AF5F18B038	11/9/25	

Exportamos los certificados del Sub_CA usando PEM CHAIN. (contiene la cadena de certificación completa)

X Certificate and Key management

Certificate export

Name Sub CA

Filename /home/kali/Desktop/SubCA.pem

Concatenated text format of the complete certificate chain in one PEM file

☐ Export comment into PEM file

Export Format PEM chain (*.pem)

OK Cancel

Y luego lo movemos a la ruta /etc/apache2/SubCA.pem

```

File Actions Edit View Help

(kali@kali)-[~/Desktop]
$ sudo mv SubCA.pem /etc/apache2
[sudo] password for kali:

(kali@kali)-[~/Desktop]
$ ls -l /etc/apache2
total 88
-rw-r--r-- 1 root root 7224 Jan 11 2021 apache2.conf
drwxr-xr-x 2 root root 4096 Apr 3 2021 conf-available
drwxr-xr-x 2 root root 4096 Apr 3 2021 conf-enabled
-rw-r--r-- 1 root root 1782 Aug 8 2020 envvars
-rw-r--r-- 1 root root 31063 Aug 8 2020 magic
drwxr-xr-x 2 root root 12288 Apr 5 2021 mods-available
drwxr-xr-x 2 root root 4096 Oct 28 20:38 mods-enabled
-rw-r--r-- 1 root root 320 Aug 8 2020 ports.conf
drwxr-xr-x 2 root root 4096 Apr 3 2021 sites-available
drwxr-xr-x 2 root root 4096 Nov 9 12:34 sites-enabled
-rw-r--r-- 1 kali kali 6313 Nov 9 13:51 SubCA.pem

```


Configuracion de apache para soportar autenticacion cliente

Entramos al archivo de configuracion de apache usando **sudo nano /etc/apache2/sites-enabled/000-default.conf** y agregamos las ultimas 6 lineas dentro del virtualHost:

```
<VirtualHost *:443>
    ServerName apache.empresa.com
    SSLEngine on
    SSLCertificateFile "/home/kali/Desktop/apache.empresa.com.crt"
    SSLCertificateKeyFile "/home/kali/Desktop/apache.key"
    DirectoryIndex index.php index.html

    ErrorLog ${APACHE_LOG_DIR}/SSLerror.log
    CustomLog ${APACHE_LOG_DIR}/SSLaccess.log combined

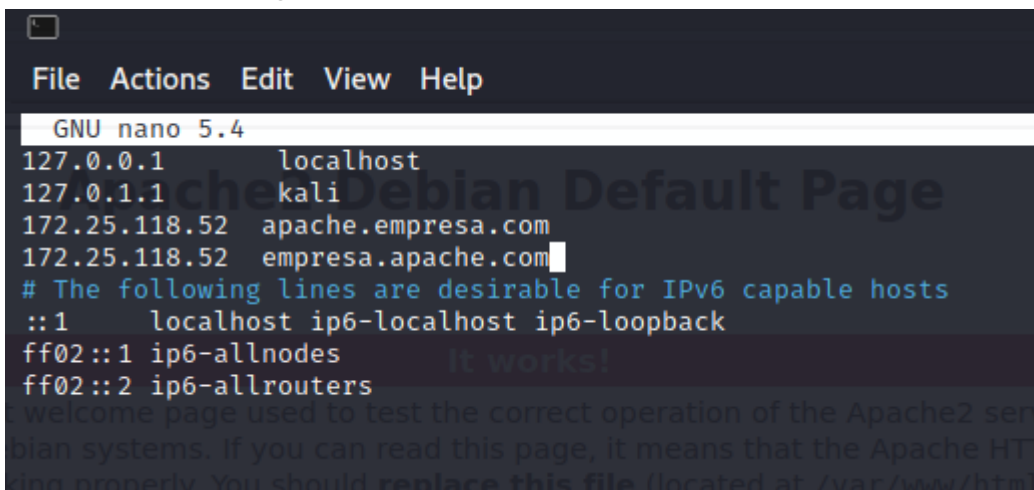
    SSLCACertificateFile "/etc/apache2/SubCA.pem"
    SSLVerifyClient require
    SSLVerifyDepth 3
<Location / >
    SSLOptions +StdEnvVars -ExportCertData
    Options FollowSymLinks
    AllowOverride None
</Location>

</VirtualHost>
```

Ahora intentamos entrar a:

<https://empresa.apache.com>

Pero antes configuramos el dns en /etc/host:



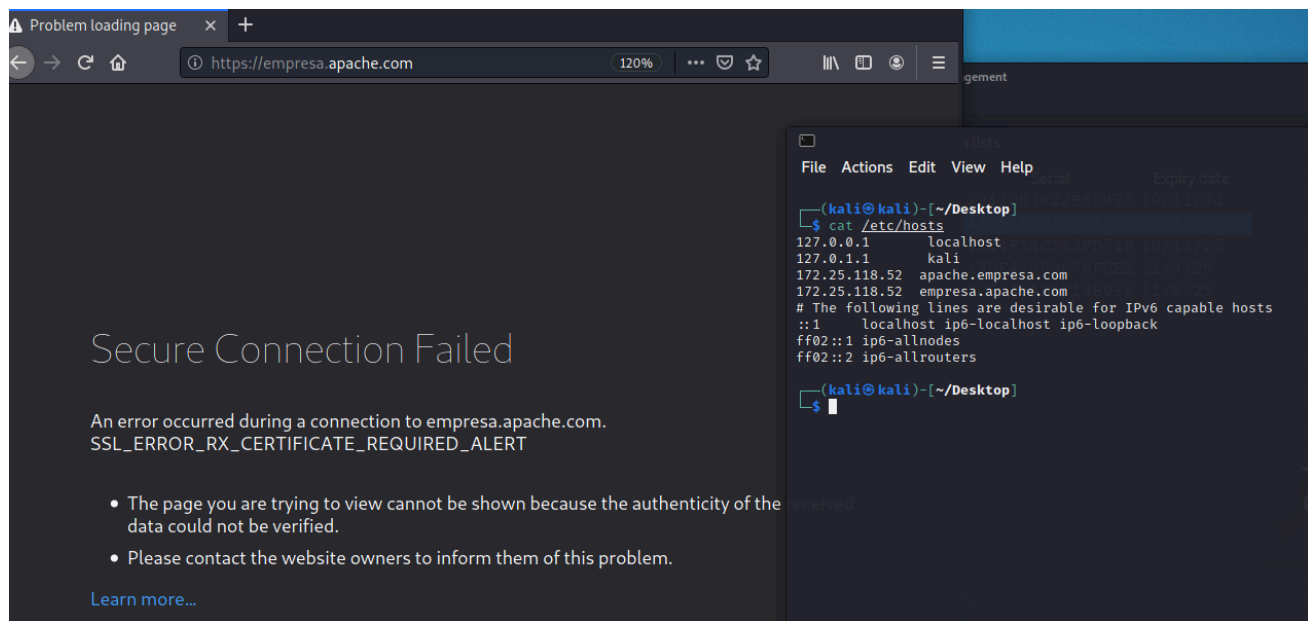
```
File Actions Edit View Help
GNU nano 5.4
127.0.0.1    localhost
127.0.1.1    kali
172.25.118.52 apache.empresa.com
172.25.118.52 empresa.apache.com
# The following lines are desirable for IPv6 capable hosts
::1        localhost ip6-localhost ip6-loopback
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters
welcome page used to test the correct operation of the Apache2 ser...
```

Observamos recordar que si modificamos el archivo de configuración de apache debemos reiniciar el apache:

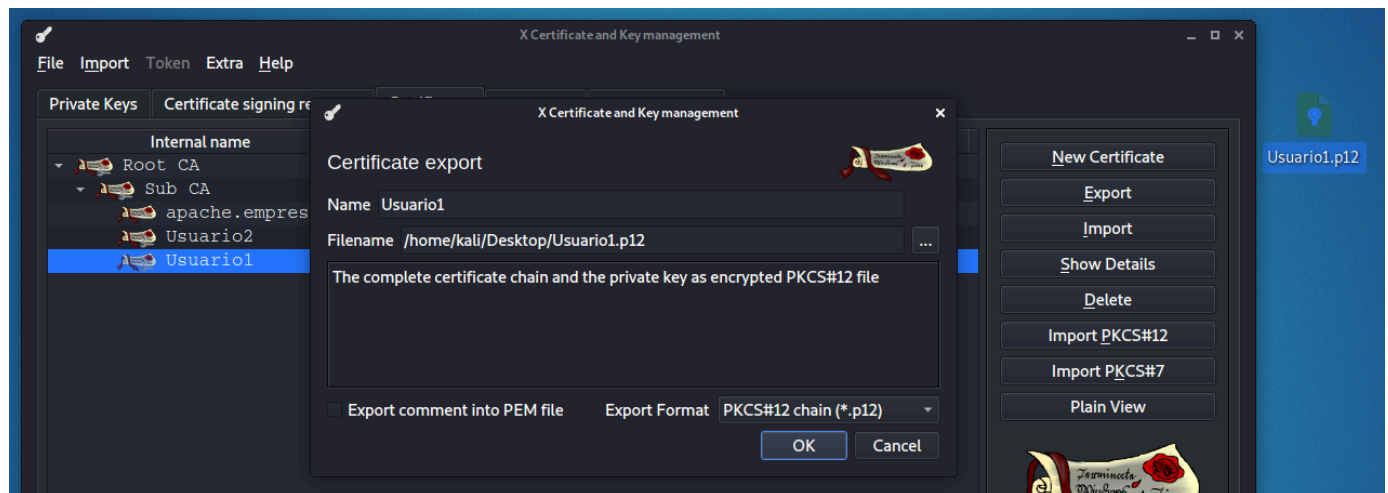
service apache2 restart

Nos dice ERROR_RX_CERTIFICAT_REQUIRED_ALERT:

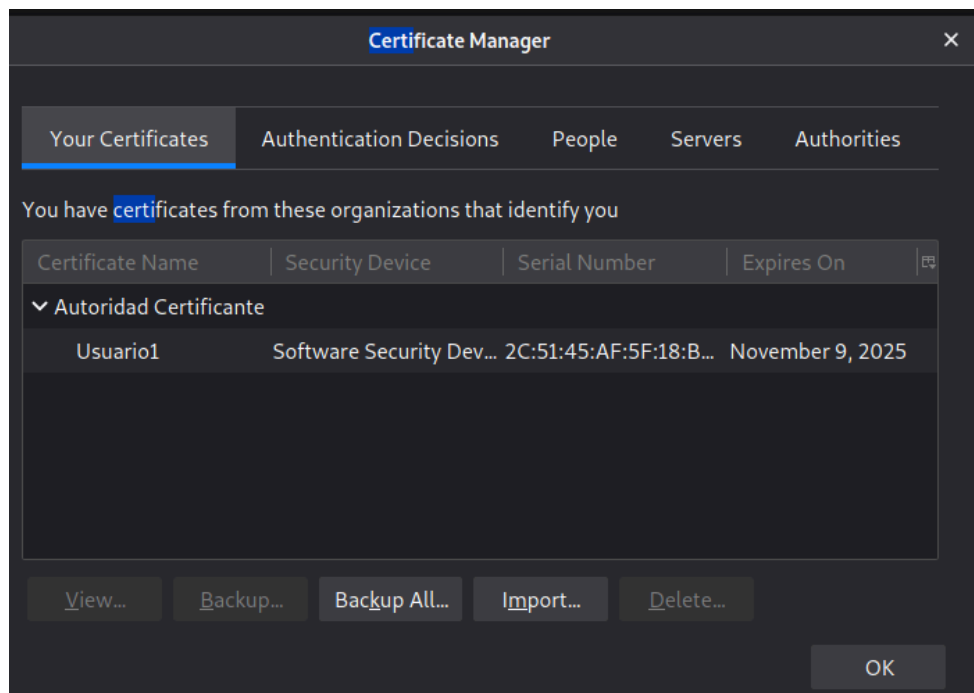
<https://empresa.apache.com>



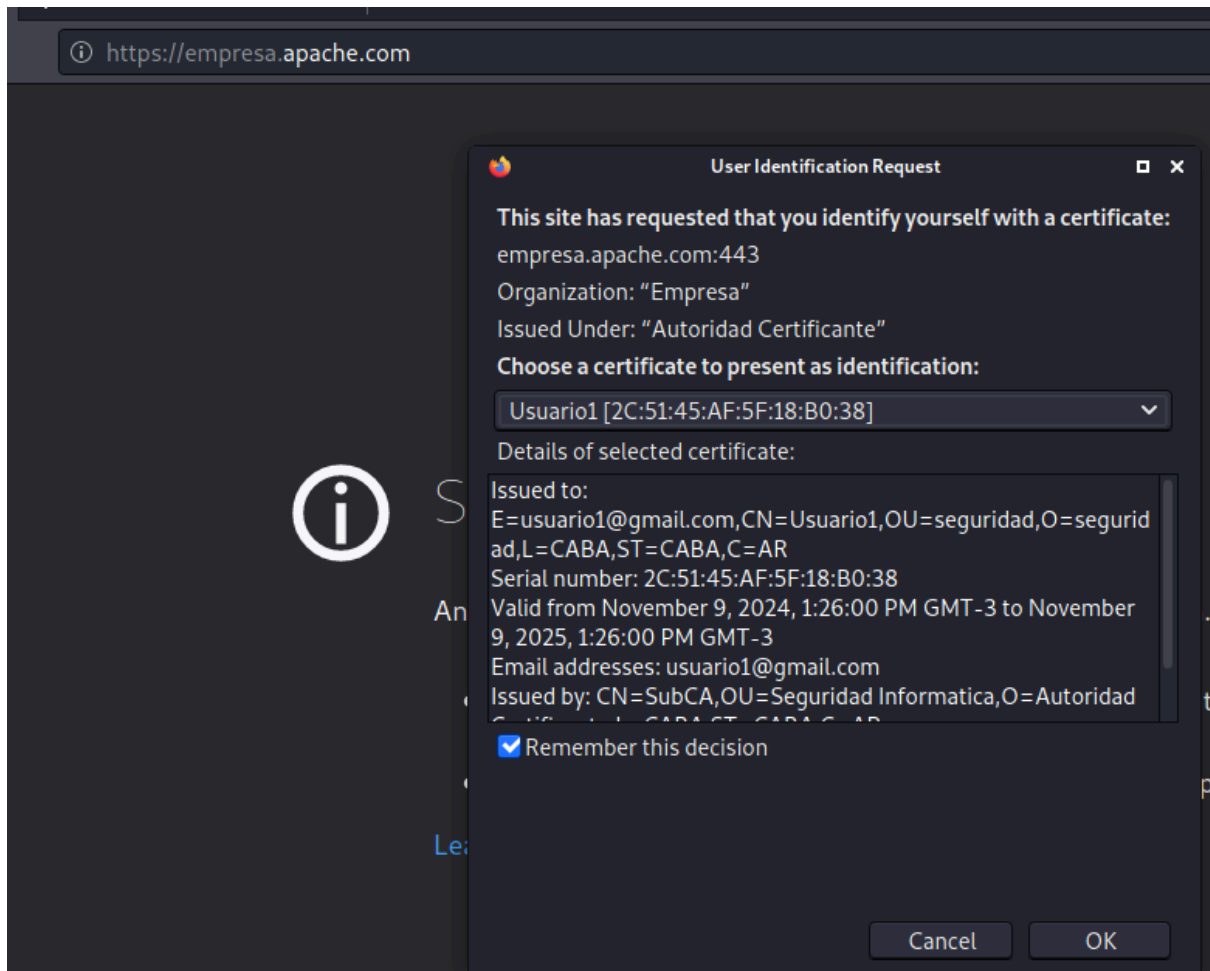
Usando el **XCA**, exportamos un .p12 para el usuario1 (con contraseña 123456).



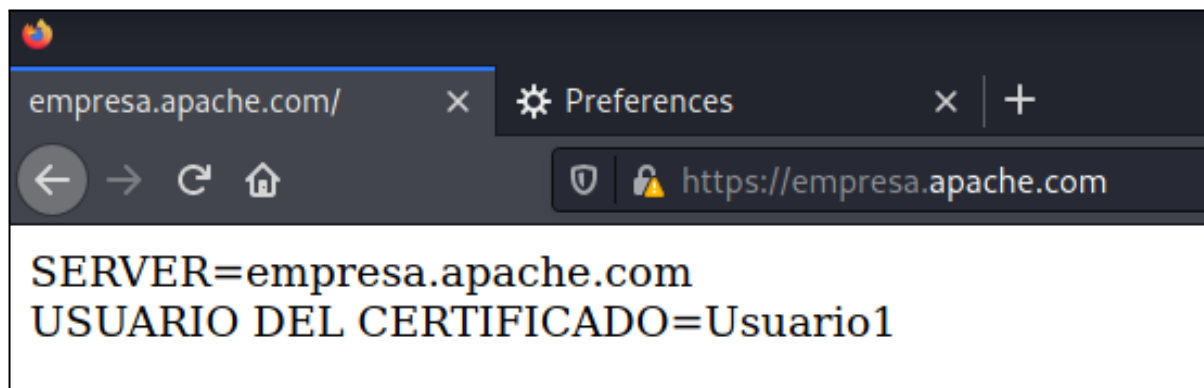
Lo importamos al firefox (le colocamos la contraseña: 123456):



Si entramos a <https://empresa.apache.com> ahora nos pide que usemos el certificado:



Aceptamos y nos aparece el nombre del usuario que le pusimos en el XCA.



Control de acceso basado en atributos del usuario

Vamos al archivo de configuración de apache y agregamos un nuevo Location con SSLOptions y SSLRequire

```

<VirtualHost *:443>
    ServerName apache.empresa.com
    SSLEngine on
    SSLCertificateFile "/home/kali/Desktop/apache.empresa.com.crt"
    SSLCertificateKeyFile "/home/kali/Desktop/apache.key"
    DirectoryIndex index.php index.html

    SSLCACertificateFile "/etc/apache2/SubCA.pem"

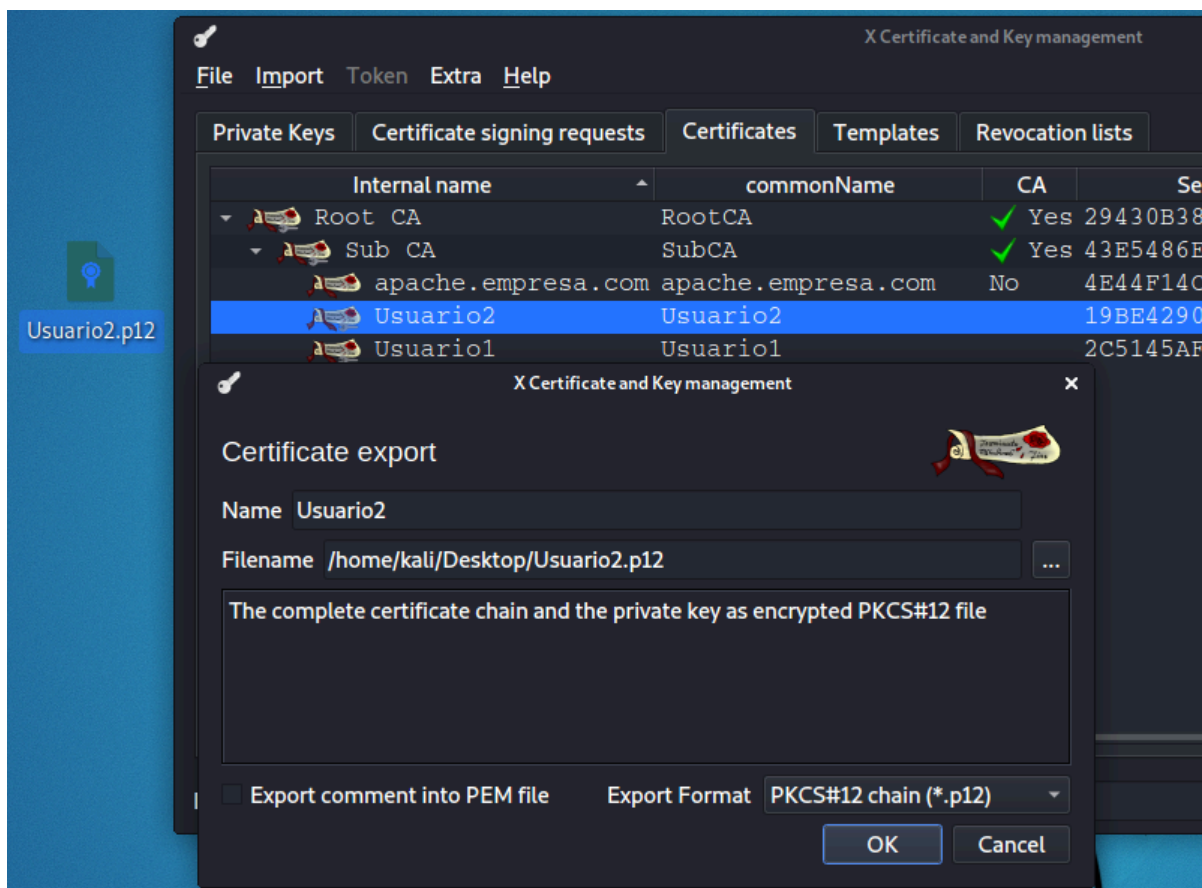
    SSLVerifyClient require
    SSLVerifyDepth 3
    <Location / >
        SSLOptions +StdEnvVars -ExportCertData
        Options FollowSymLinks
        AllowOverride None
    </Location>
    <Location /seguro.php >
        SSLOptions +StdEnvVars -ExportCertData
        SSLRequire %{SSL_CLIENT_S_DN_O} in {"contable_org", "desarrollo"}
        Options FollowSymLinks
        AllowOverride None
    </Location>
    ErrorLog ${APACHE_LOG_DIR}/SSLerror.log
    CustomLog ${APACHE_LOG_DIR}/SSLaccess.log combined
</VirtualHost>

```

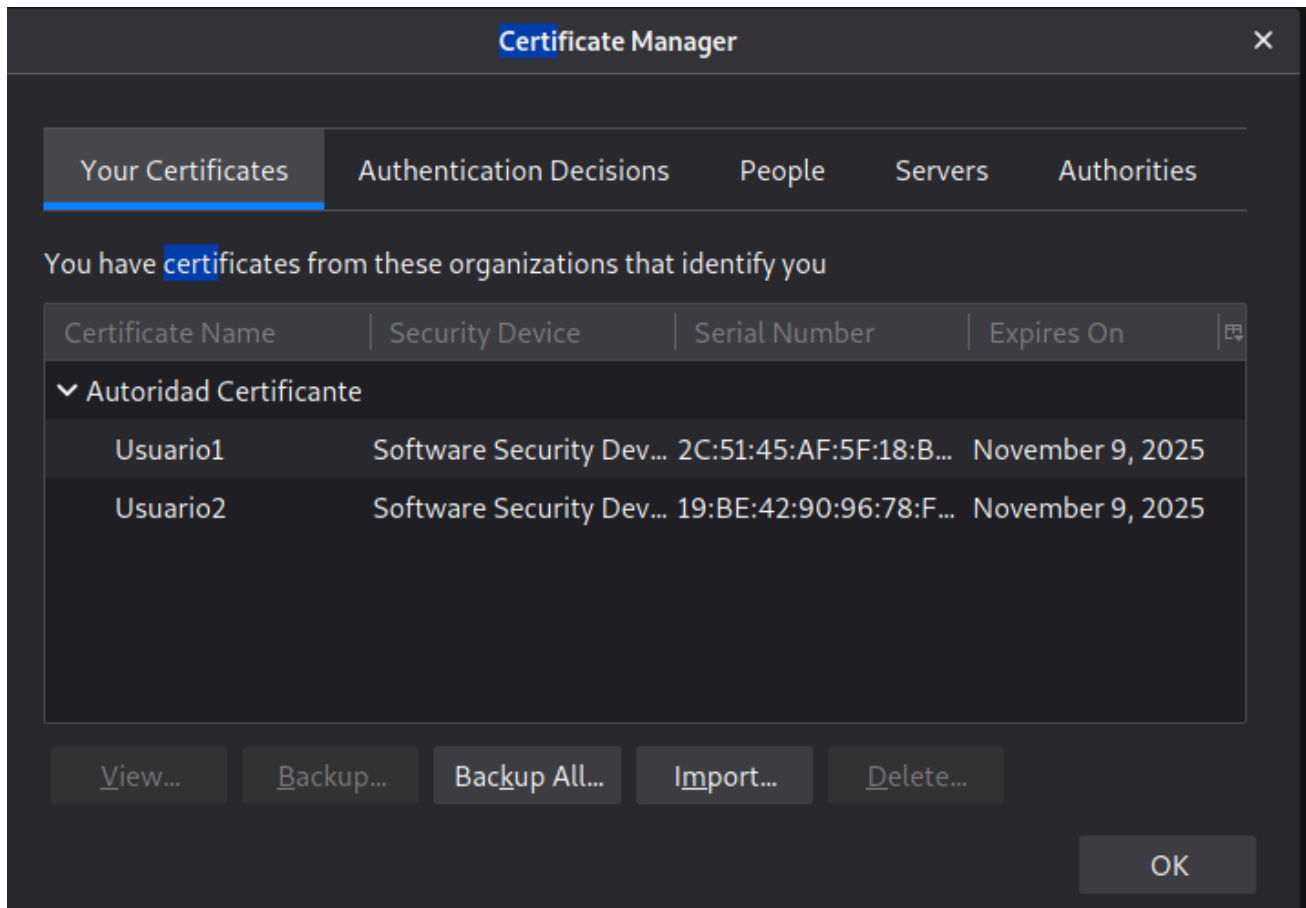
(Observacion: en el require va todo el {} completo y no abajo) (Asi evitamos error de sintaxis).

Reiniciamos el apache.

Exportamos el .p12 del usuario2 usando la cadena de certificación completa.
(contraseña: 123456)



Lo importamos a firefox (le ponemos contraseña 123456):



Ahora si navegamos a:

<https://apache.empresa.com/seguro.php>

Si entramos con el usuario 1:



Si entramos con el usuario 2 (que esta permitido porque pertenece a la organización contable_org. q esta en el archivo config de apache.



3) ¿que contiene la variable de PHP SSL_PROTOCOL?

Se observa que contiene la versión de TLS utilizado en la conexión actual. En este caso es **TLSv1.3**.

4) ¿que contiene la variable de PHP SSL_CIPHER?

Nos muestra el cifrado específico que se está utilizando en la conexión SSL/TLS en este caso es **TLS_AES_128_GCM_SHA256**.

5) ¿que contiene la variable de PHP SSL_CLIENT_I_DN_CN?

Contiene una serie de variables con el mismo estandar de nombres que el RFC 2253. Cada acronimo es una parte del nombre distinguido del emisor certificante:

CN: Common Name: Es el nombre del emisor certificante en este caso **SubCA**.

OU: Unidad organizativa. (Seguridad informatica)

O(Organizacion): nombre de la autoridad certificante.

L:Locality , **C:**Country, **ST:**State.

```
apache.empresa.com/seguri X +
https://apache.empresa.com/seguro.php
SSL_CLIENT_I_DN_O=>Autoridad Certificante
SSL_CLIENT_I_DN_OU=>Seguridad Informatica
SSL_CLIENT_I_DN_CN=>SubCA
SSL_SERVER_SAN_DNS_0=>apache.empresa.com
SSL_SERVER_SAN_DNS_1=>prueba.empresa.com
SSL_VERSION_INTERFACE=>mod_ssl/2.4.46
SSL_VERSION_LIBRARY=>OpenSSL/1.1.1k
SSL_PROTOCOL=>TLSv1.3
SSL_SECURE_RENEG=>>false
SSL_COMPRESS_METHOD=>NULL
SSL_CIPHER=>TLS_AES_128_GCM_SHA256
SSL_CIPHER_EXPORT=>>false
SSL_CIPHER_USEKEYSIZE=>128
SSL_CIPHER_ALGKEYSIZE=>128
SSL_CLIENT_VERIFY=>SUCCESS
SSL_CLIENT_M_VERSION=>3
SSL_CLIENT_M_SERIAL=>19BE42909678FCE2
SSL_CLIENT_V_START=>Nov 9 16:19:00 2024 GMT
SSL_CLIENT_V_END=>Nov 9 16:19:00 2025 GMT
SSL_CLIENT_V_REMAIN=>365
SSL_CLIENT_S_DN=>emailAddress=usuario2@gmail.com,CN=Usuario2,OU=contable_org,O=contable_org,L=CABA/
SSL_CLIENT_I_DN=>CN=SubCA,OU=Seguridad Informatica,O=Autoridad Certificante,L=CABA,ST=CABA,C=AR
```

Sub CA

SubCA

Yes 43E5486E13B84FAD 10/11/29

apache.empresa.com

apache.empresa.com

No 4E44F14C113FD748 10/11/25

X Certificate and Key management

Details of the Certificate

Status	Subject	Issuer	Extensions	Comment
	countryName	AR		
	stateOrProvinceName	CABA		
	localityName	CABA		
	organizationName	Autoridad Certificante		
	organizationalUnitName	Seguridad Informatica		
	commonName	RootCA		

RFC 2253: CN=RootCA,OU=Seguridad Informatica,O=Autoridad Certificante,L=CABA,ST=CABA,C=AR

Hash: 6c037c96

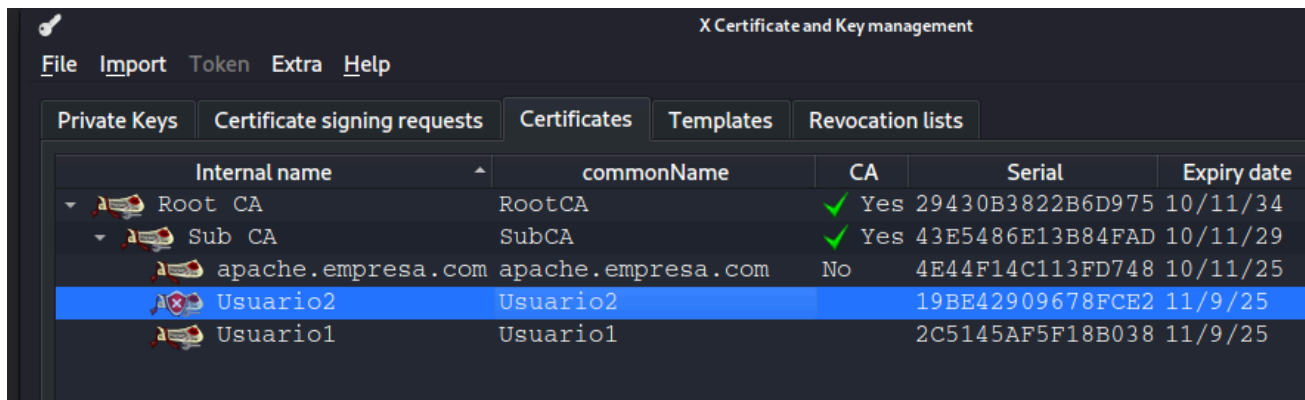
Si revisamos el Sub_CA en el XCA veremos que la misma convención (RFC 2253).

Opcionales

Revocación de Certificado y CRL

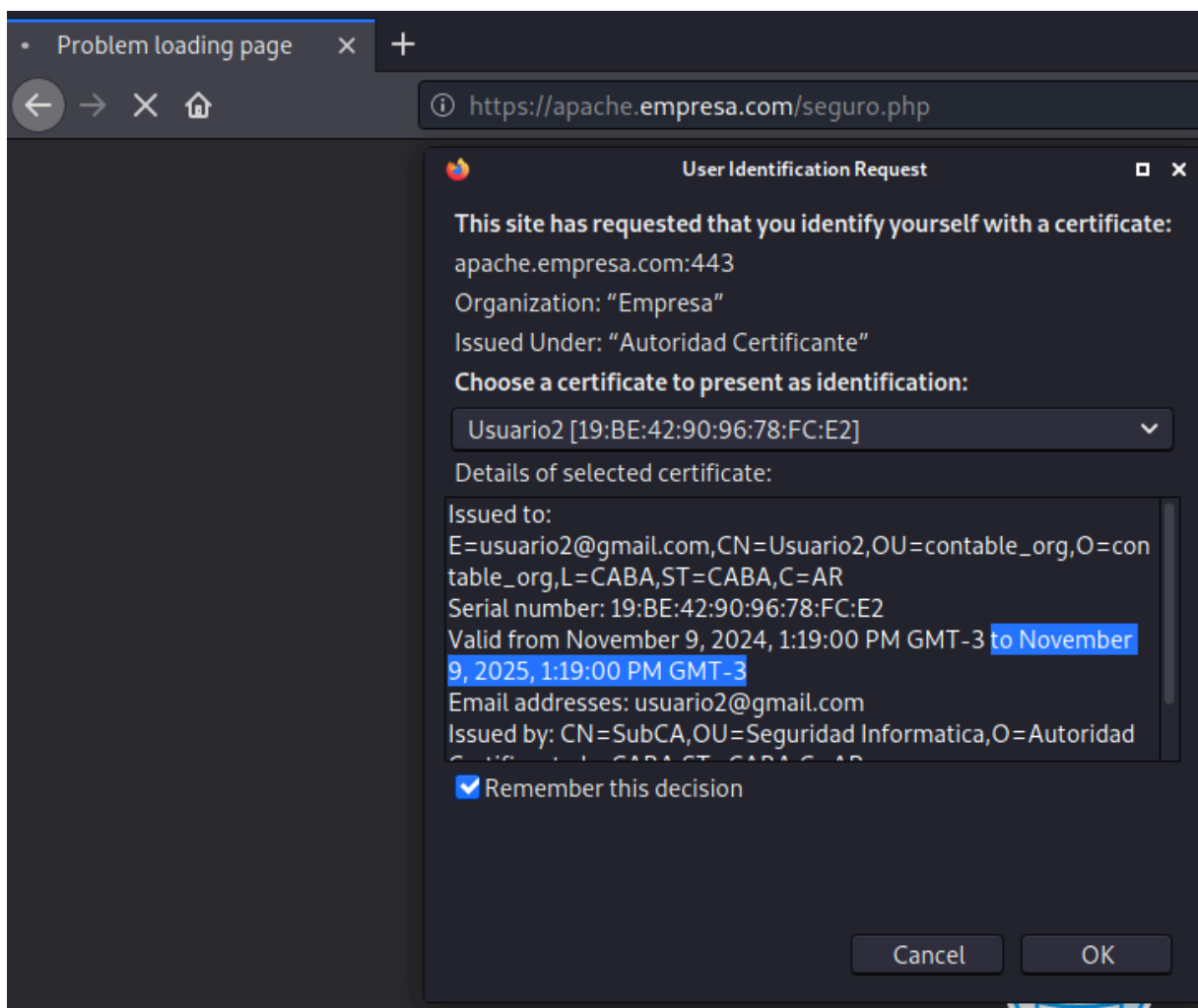
1) Entrar a la xca y revocar el certificado de Usuario2, ver que aun puede entrar al apache. ¿por qué?

Revocamos al usuario2 en el XCA:



Internal name	commonName	CA	Serial	Expiry date
Root CA	RootCA	✓ Yes	29430B3822B6D975	10/11/34
Sub CA	SubCA	✓ Yes	43E5486E13B84FAD	10/11/29
apache.empresa.com	apache.empresa.com	No	4E44F14C113FD748	10/11/25
Usuario2	Usuario2		19BE42909678FCE2	11/9/25
Usuario1	Usuario1		2C5145AF5F18B038	11/9/25

Podemos seguir entrando a seguro.ph porque estamos usando el .p12 antiguo cuya validacion es hasta 9 Nov 2025. Apache aun no tiene conocimiento que se revoco ese certificado.



Podemos acceder a la info aun cuando el usuario2 tiene el certificado revocado.

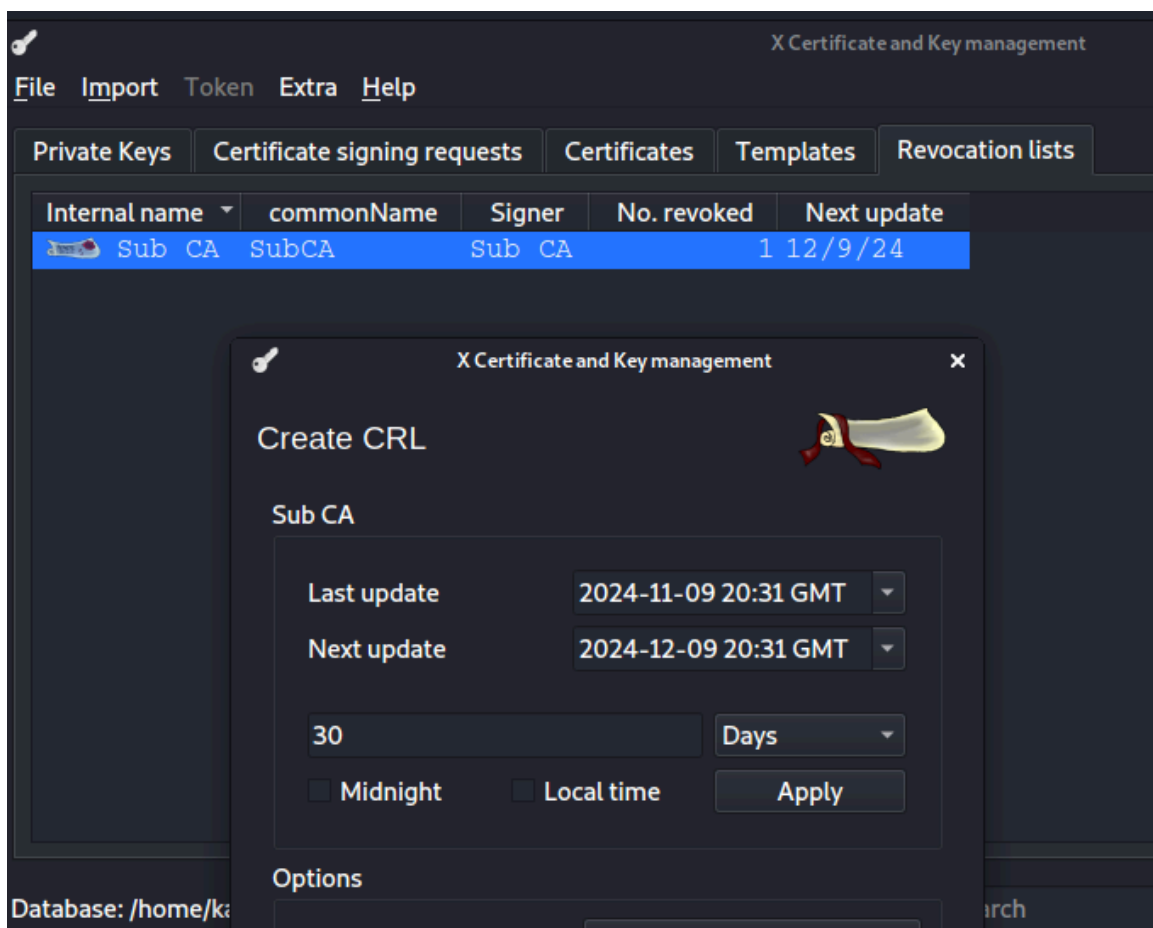


2) en la xca generar una CRL y exportarla como

/home/kali/SubCACrl.pem

luego poner en el archivo de configuración de apache las siguientes directivas:

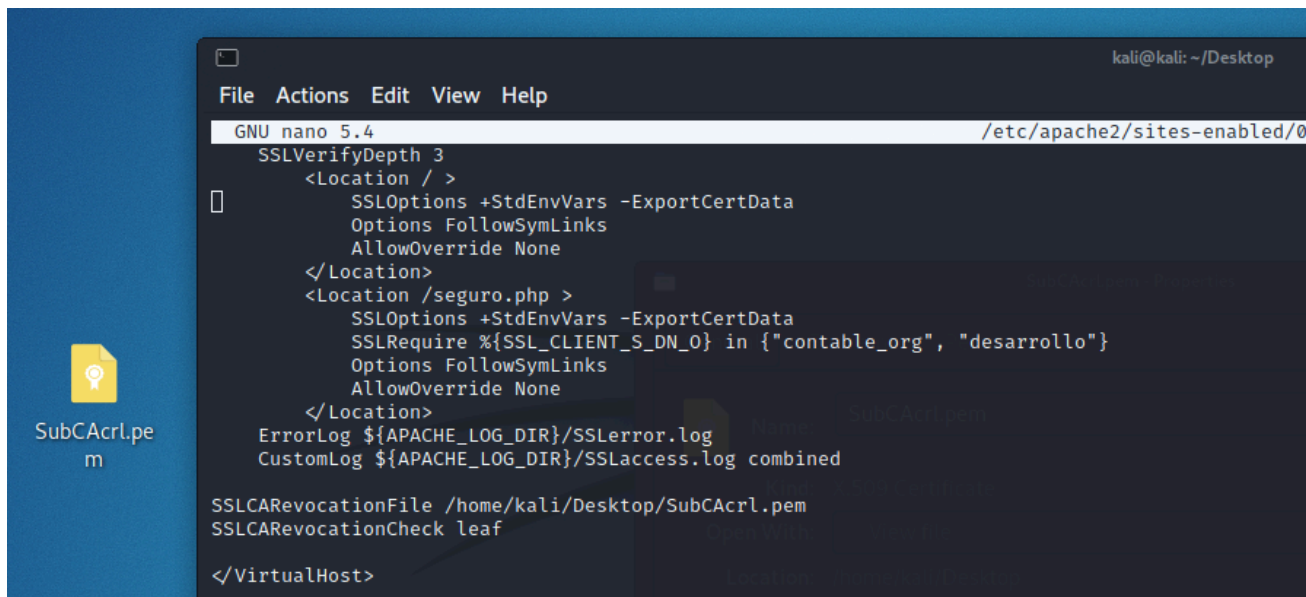
Creamos el CRL Certificate revocation list.



Lo guardo en desktop y luego modificamos el archivo de configuracion de apache:

SSLCARevocationFile /home/kali/Desktop/SubCAcrl.pem

SSLCARevocationCheck leaf



```
File Actions Edit View Help
GNU nano 5.4 /etc/apache2/sites-enabled/0
SSLVerifyDepth 3
<Location / >
  SSLOptions +StdEnvVars -ExportCertData
  Options FollowSymLinks
  AllowOverride None
</Location>
<Location /seguro.php >
  SSLOptions +StdEnvVars -ExportCertData
  SSLRequire %{SSL_CLIENT_S_DN_O} in {"contable_org", "desarrollo"}
  Options FollowSymLinks
  AllowOverride None
</Location>
ErrorLog ${APACHE_LOG_DIR}/SSLerror.log
CustomLog ${APACHE_LOG_DIR}/SSLaccess.log combined

SSLCARevocationFile /home/kali/Desktop/SubCAcrl.pem
SSLCARevocationCheck leaf

</VirtualHost>
```

Ahora reiniciamos el apache y entramos con el certificado de user2:

A pesar de que diga: "Valid from November 9, 2024, 1:19:00 PM GMT-3 to November 9, 2025, 1:19:00 PM GMT-3"

Ya no nos deja entrar, porque ahora apache tiene conocimiento de que se revocó ese certificado.

