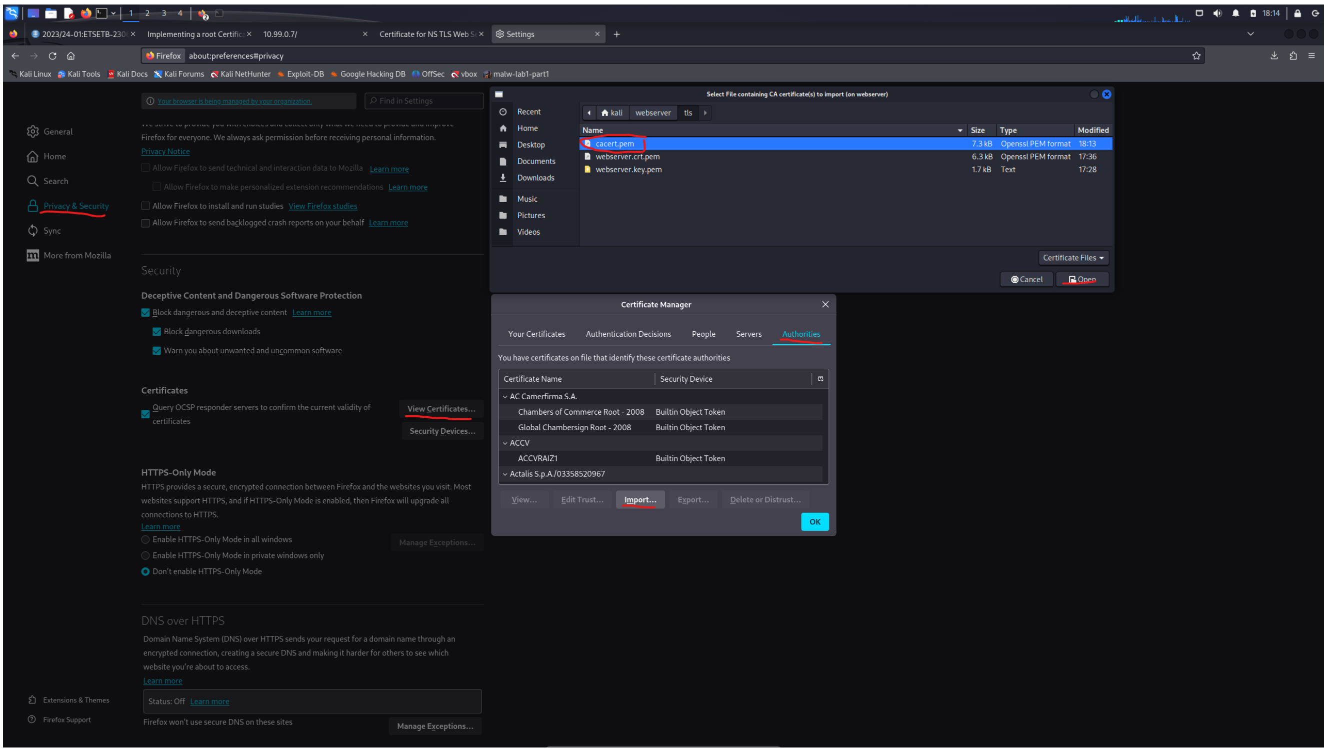
[NS - Lab2] - Implementing a root Certification Authority with OpenSSL

3.1 Test your TLS webserver with a web browser

3.1.1 - Install the CA's certificate on the client's web browser



Your browser is being managed by your organization.

Firefox for everyone. We always ask permission before receiving personal information.
Privacy Notice

□ Allow Firefox to send technical and interaction data to Mozilla Learn more

□ Allow Firefox to make personalized extension recommendations Learn more

□ Allow Firefox to install and run studies View Firefox studies

□ Allow Firefox to send backlogged crash reports on your behalf Learn more

Security

Deceptive Content and Dangerous Software Protection

☑ Block dangerous and deceptive content Learn more

☑ Block dangerous downloads

☑ Warn you about unwanted and uncommon software

Certificates

☑ Query OCSP responder servers to confirm the current validity of certificates

View Certificates…

Security Devices…

HTTPS-Only Mode

HTTPS provides a secure, encrypted connection between Firefox and the websites you visit. Most websites support HTTPS, and if HTTPS-Only Mode is enabled, then Firefox will upgrade all connections to HTTPS.
Learn more

○ Enable HTTPS-Only Mode in all windows

○ Enable HTTPS-Only Mode in private windows only

● Don't enable HTTPS-Only Mode

Manage Exceptions…

DNS over HTTPS

Domain Name System (DNS) over HTTPS sends your request for a domain name through an encrypted connection, creating a secure DNS and making it harder for others to see which website you're about to access.
Learn more

Status: Off Learn more

Firefox won't use secure DNS on these sites

Manage Exceptions…

Select File containing CA certificate(s) to import (on webserver)

kali / webserver / tls

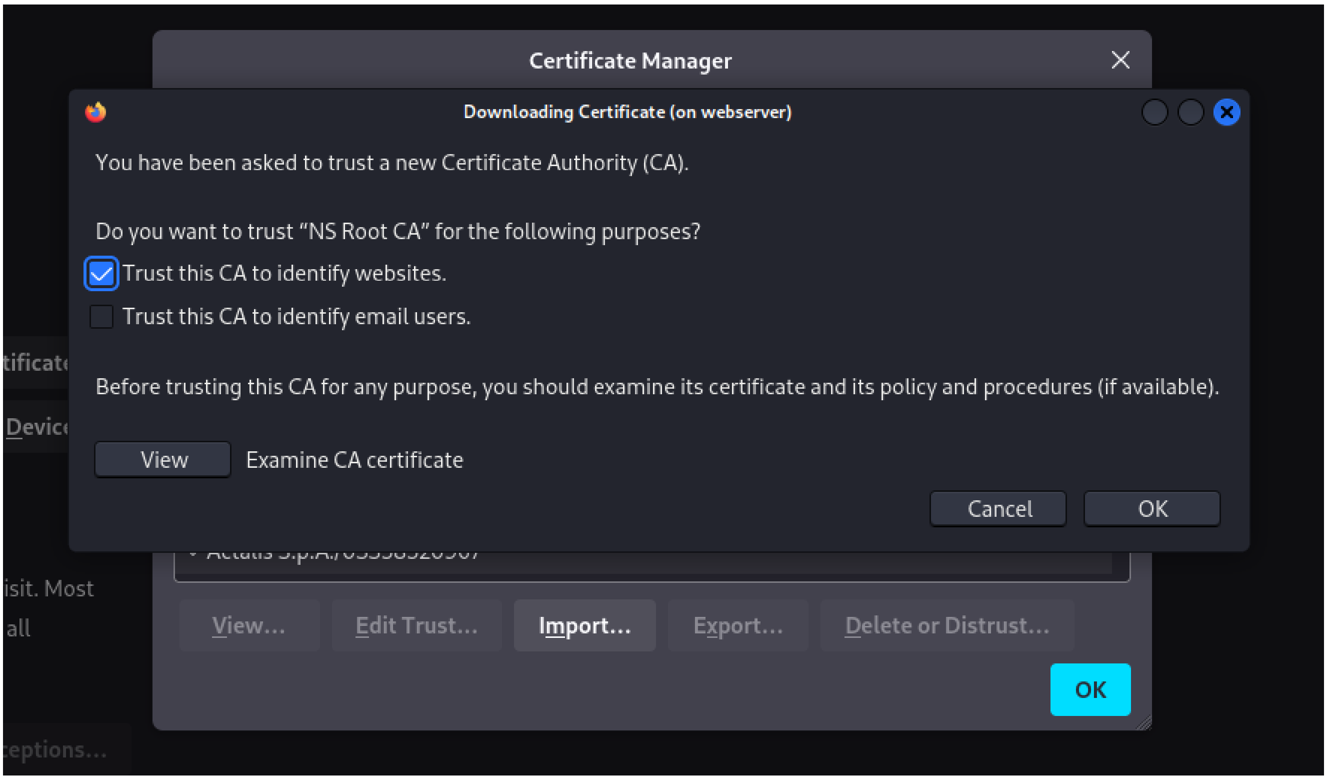
Name	Size	Type	Modified
cacert.pem	7.3 kB	Openssl PEM format	18:13
webserver.crt.pem	6.3 kB	Openssl PEM format	17:36
webserver.key.pem	1.7 kB	Text	17:28

Certificate Manager

You have certificates on file that identify these certificate authorities

Certificate Name	Security Device
AC Camerfirma S.A.	
Chambers of Commerce Root - 2008	Builtin Object Token
Global Chambersign Root - 2008	Builtin Object Token
ACCV	
ACCVRAIZ1	Builtin Object Token
Actalis S.p.A./03358520967	

Image 1



Downloading Certificate (on webserver)

You have been asked to trust a new Certificate Authority (CA).

Do you want to trust "NS Root CA" for the following purposes?

☑ Trust this CA to identify websites.

□ Trust this CA to identify email users.

Before trusting this CA for any purpose, you should examine its certificate and its policy and procedures (if available).

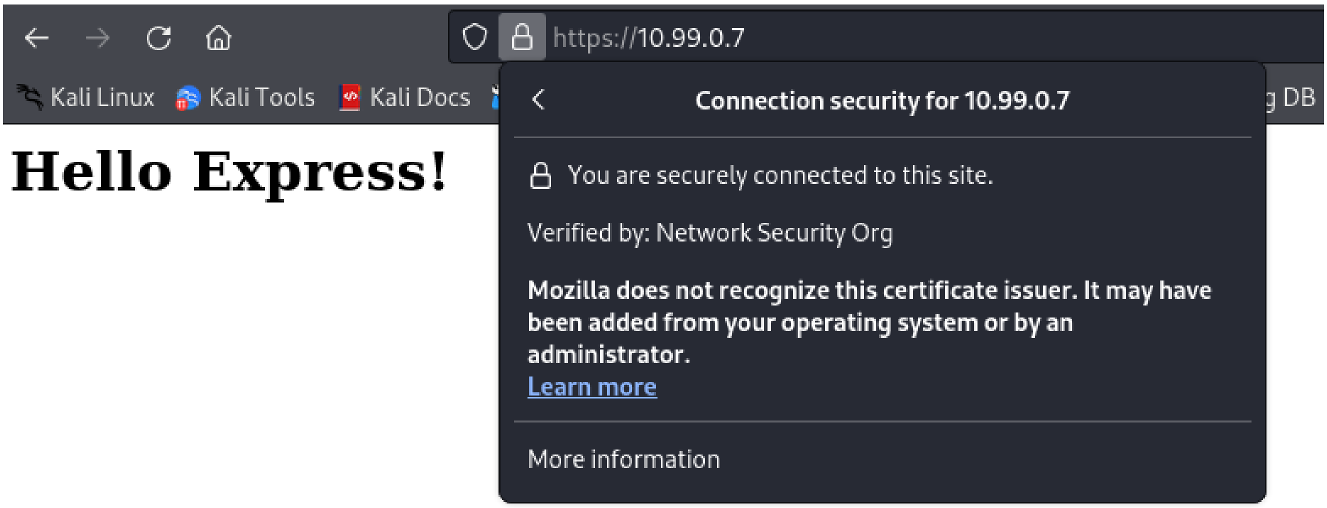
View Examine CA certificate

Cancel OK

Image 2

Here we just go to our web browser (Firefox in this case) and search and install for our new root CA certificate.

3.1.2 - Connect the browser to the server https://10.0.2.7



Hello Express!

Connection security for 10.99.0.7

You are securely connected to this site.

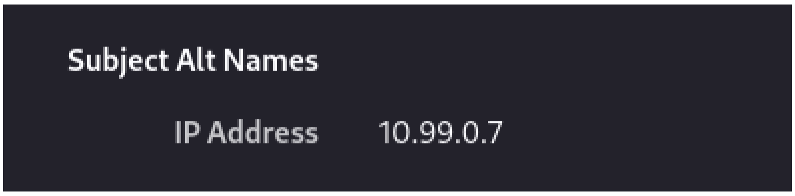
Verified by: Network Security Org

Mozilla does not recognize this certificate issuer. It may have been added from your operating system or by an administrator.

Image 3

Certificate seems fine! Let's keep going.

3.1.3 - Check that the connection does not raise any warnings (unless the server’s certificate had an inappropriate subjectAltName extension)



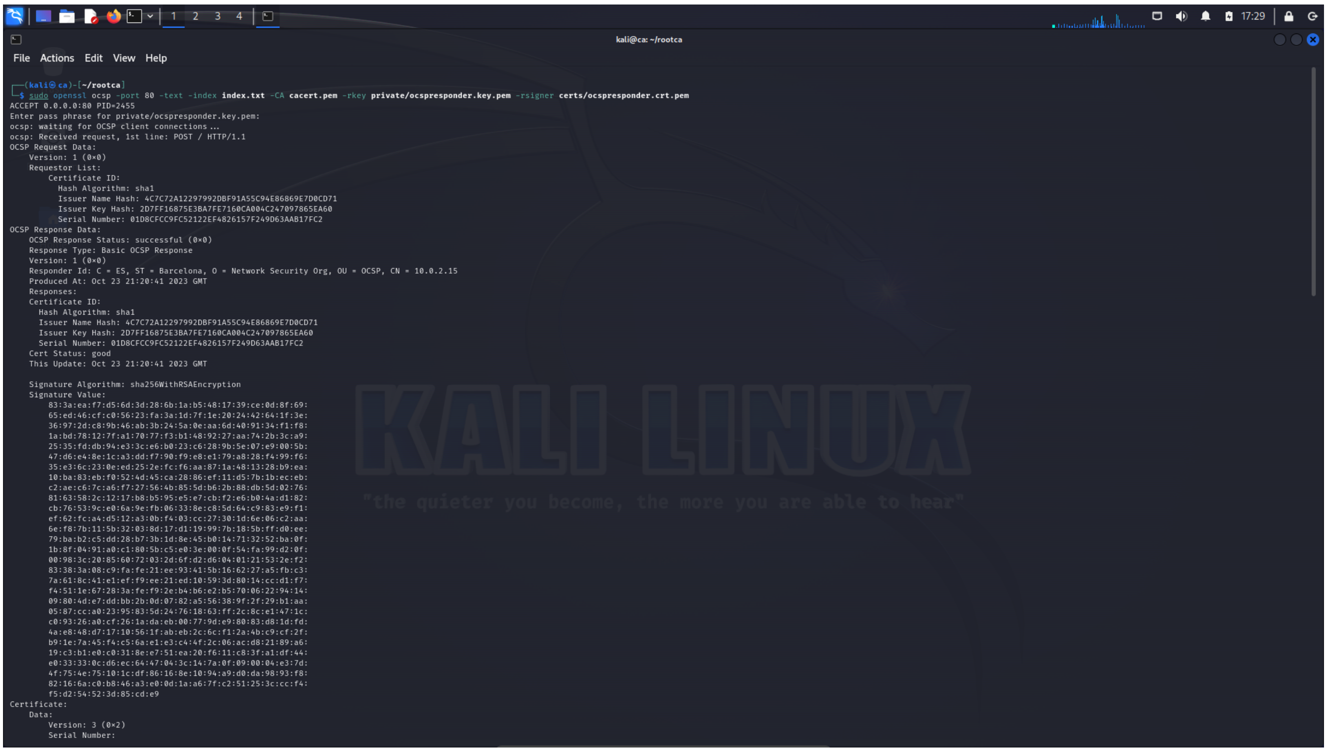
Subject Alt Names

IP Address 10.99.0.7

Image 4

Alt name properly configured.

3.1.4 - Check also that the web browser has also queried your OCSP responder to check the status of the server's certificate

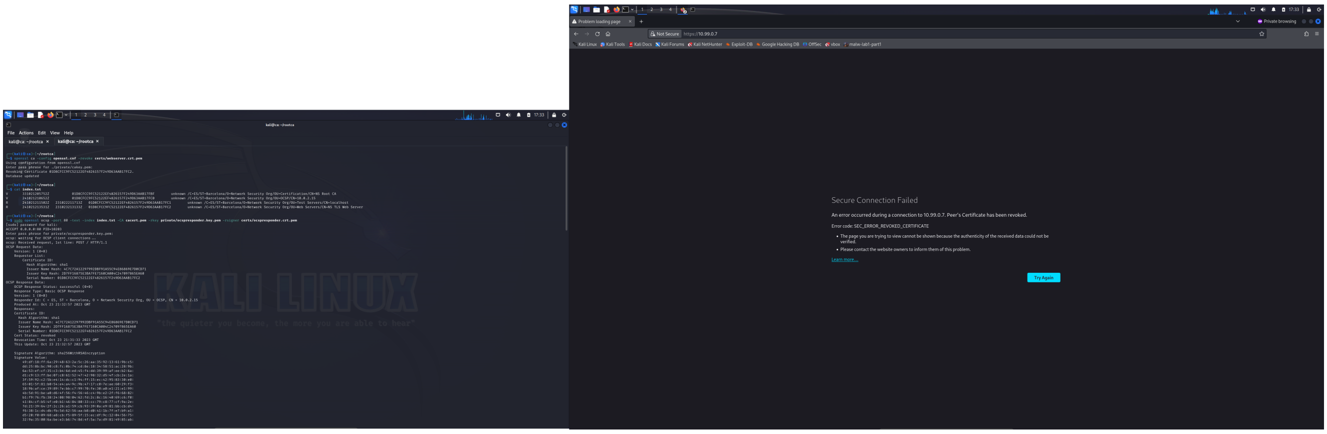


```
┌──(kali㉿ca)-[~/rootca]
└─$ sudo openssl ocsp -port 80 -text -index index.txt -CA cacert.pem -rkey private/ocspresponder.key.pem -rsigner certs/ocspresponder.crt.pem
ACCEPT 0.0.0.0:80 PID=2455
Enter pass phrase for private/ocspresponder.key.pem:
ocsp: waiting for OCSP client connections ...
ocsp: Received request, 1st line: POST / HTTP/1.1
OCSP Request Data:
    Version: 1 (0×0)
    Requestor List:
        Certificate ID:
          Hash Algorithm: sha1
          Issuer Name Hash: 4C7C72A12297992DBF91A55C94E86869E7D0CD71
          Issuer Key Hash: 2D7FF16875E3BA7FE7160CA004C247097865EA60
          Serial Number: 01D8CFCC9FC52122EF4826157F249D63AAB17FC2
OCSP Response Data:
    OCSP Response Status: successful (0×0)
    Response Type: Basic OCSP Response
    Version: 1 (0×0)
    Responder Id: C = ES, ST = Barcelona, O = Network Security Org, OU = OCSP, CN = 10.0.2.15
    Produced At: Oct 23 21:20:41 2023 GMT
    Responses:
    Certificate ID:
      Hash Algorithm: sha1
      Issuer Name Hash: 4C7C72A12297992DBF91A55C94E86869E7D0CD71
      Issuer Key Hash: 2D7FF16875E3BA7FE7160CA004C247097865EA60
      Serial Number: 01D8CFCC9FC52122EF4826157F249D63AAB17FC2
    Cert Status: good
    This Update: Oct 23 21:20:41 2023 GMT

    Signature Algorithm: sha256WithRSAEncryption
    Signature Value:
        83:3a:ea:f7:d5:6d:3d:28:6b:1a:b5:48:17:39:ce:0d:8f:69:
        65:ed:46:cf:c0:56:23:fa:3a:1d:7f:1e:20:24:42:64:1f:3e:
        36:97:2d:c8:9b:46:ab:3b:24:5a:0e:aa:6d:40:91:34:f1:f8:
        1a:bd:78:12:7f:a1:70:77:f3:b1:48:92:27:aa:74:2b:3c:a9:
        25:35:fd:db:94:e3:3c:e6:b0:23:c6:28:9b:5e:07:e9:00:5b:
        47:d6:e4:8e:1c:a3:dd:f7:90:f9:e8:e1:79:a8:28:f4:99:f6:
        35:e3:6c:23:0e:ed:25:2e:fc:f6:aa:87:1a:48:13:28:b9:ea:
        10:ba:83:eb:f0:52:4d:45:ca:28:86:ef:11:d5:7b:1b:ec:eb:
        c2:ae:c6:7c:a6:f7:27:56:4b:85:5d:b6:2b:88:db:5d:02:76:
        81:63:58:2c:12:17:b8:b5:95:e5:e7:cb:f2:e6:b0:4a:d1:82:
        cb:76:53:9c:e0:6a:9e:fb:06:33:8e:c8:5d:64:c9:83:e9:f1:
        ef:62:fc:a4:d5:12:a3:0b:f4:03:cc:27:30:1d:6e:06:c2:aa:
        6e:f8:7b:11:5b:32:03:8d:17:d1:19:99:7b:18:5b:ff:d0:ee:
        79:ba:b2:c5:dd:28:b7:3b:1d:8e:45:b0:14:71:32:52:ba:0f:
        1b:8f:04:91:a0:c1:80:5b:c5:e0:3e:00:0f:54:fa:99:d2:0f:
        00:98:3c:20:85:60:72:03:2d:6f:d2:d6:04:01:21:53:2e:f2:
        83:38:3a:08:c9:fa:fe:21:ee:93:41:5b:16:62:27:a5:fb:c3:
        7a:61:8c:41:e1:ef:f9:ee:21:ed:10:59:3d:80:14:cc:d1:f7:
        f4:51:1e:67:28:3a:fe:f9:2e:b4:b6:e2:b5:70:06:22:94:14:
        09:80:4d:e7:dd:bb:2b:0d:07:82:a5:56:38:9f:2f:29:b1:aa:
        05:87:cc:a0:23:95:83:5d:24:76:18:63:ff:2c:8c:e1:47:1c:
        c0:93:26:a0:cf:26:1a:da:eb:00:77:9d:e9:80:83:d8:1d:fd:
        4a:e8:48:d7:17:10:56:1f:ab:eb:2c:6c:f1:2a:4b:c9:cf:2f:
        b9:1e:7a:45:f4:c5:6a:e1:e3:c4:4f:2c:06:ac:d8:21:89:a6:
        19:c3:b1:e0:c0:31:8e:e7:51:ea:20:f6:11:c8:3f:a1:df:44:
        e0:33:33:0c:d6:ec:64:47:04:3c:14:7a:0f:09:00:04:e3:7d:
        4f:75:4e:75:10:1c:df:86:16:8e:10:94:a9:d0:da:98:93:f8:
        82:16:6a:c0:b8:46:a3:e0:0d:1a:a6:7f:c2:51:25:3c:cc:f4:
        f5:d2:54:52:3d:85:cd:e9
Certificate:
    Data:
        Version: 3 (0×2)
        Serial Number:
```

Image 5

OCSP responder works properly.

3.1.5 - Revoke the webserver's certificate and check that the browser is not letting you in since the certificate is revoked



```
┌──(kali㉿ca)-[~/rootca]
└─$ openssl ca -config openssl.cnf -revoke certs/webserver.crt.pem
Using configuration from openssl.cnf
Enter pass phrase for ./private/cakey.pem:
Revoking Certificate 01D8CFCC9FC52122EF4826157F249D63AAB17FC2.
Database updated

┌──(kali㉿ca)-[~/rootca]
└─$ cat index.txt
V       331021205752Z           01D8CFCC9FC52122EF4826157F249D63AAB17FBF        unknown /C=ES/ST=Barcelona/O=Network Security Org/OU=Certification/CN=NS Root CA
V       241021210652Z           01D8CFCC9FC52122EF4826157F249D63AAB17FC0        unknown /C=ES/ST=Barcelona/O=Network Security Org/OU=OCSP/CN=10.0.2.15
R       241021211502Z   231022211713Z   01D8CFCC9FC52122EF4826157F249D63AAB17FC1        unknown /C=ES/ST=Barcelona/O=Network Security Org/OU=Test Servers/CN=localhost
R       241021213531Z   231023213133Z   01D8CFCC9FC52122EF4826157F249D63AAB17FC2        unknown /C=ES/ST=Barcelona/O=Network Security Org/OU=Web Servers/CN=NS TLS Web Server

┌──(kali㉿ca)-[~/rootca]
└─$ sudo openssl ocsp -port 80 -text -index index.txt -CA cacert.pem -rkey private/ocspresponder.key.pem -rsigner certs/ocspresponder.crt.pem
[sudo] password for kali:
ACCEPT 0.0.0.0:80 PID=10203
Enter pass phrase for private/ocspresponder.key.pem:
ocsp: waiting for OCSP client connections ...
ocsp: Received request, 1st line: POST / HTTP/1.1
OCSP Request Data:
    Version: 1 (0×0)
    Requestor List:
        Certificate ID:
          Hash Algorithm: sha1
          Issuer Name Hash: 4C7C72A12297992DBF91A55C94E86869E7D0CD71
          Issuer Key Hash: 2D7FF16875E3BA7FE7160CA004C247097865EA60
          Serial Number: 01D8CFCC9FC52122EF4826157F249D63AAB17FC2
OCSP Response Data:
    OCSP Response Status: successful (0×0)
    Response Type: Basic OCSP Response
    Version: 1 (0×0)
    Responder Id: C = ES, ST = Barcelona, O = Network Security Org, OU = OCSP, CN = 10.0.2.15
    Produced At: Oct 23 21:32:57 2023 GMT
    Responses:
    Certificate ID:
      Hash Algorithm: sha1
      Issuer Name Hash: 4C7C72A12297992DBF91A55C94E86869E7D0CD71
      Issuer Key Hash: 2D7FF16875E3BA7FE7160CA004C247097865EA60
      Serial Number: 01D8CFCC9FC52122EF4826157F249D63AAB17FC2
    Cert Status: revoked
    Revocation Time: Oct 23 21:31:33 2023 GMT
    This Update: Oct 23 21:32:57 2023 GMT

    Signature Algorithm: sha256WithRSAEncryption
    Signature Value:
        49:df:18:ff:6a:29:48:63:2a:5c:26:aa:35:92:13:61:9b:c5:
        dd:25:8b:bc:90:c8:fc:0b:74:cd:8e:18:34:58:51:ac:28:9b:
        6a:53:ef:cf:35:c3:b4:6d:ed:45:f4:dd:39:99:af:ee:b2:6a:
        d1:c9:13:ff:be:07:c8:61:52:47:42:98:32:d5:4f:cb:2e:1a:
        3f:59:92:c2:5b:e4:14:dc:c1:94:ff:15:ec:42:95:83:30:e0:
        65:01:5f:81:b0:54:e4:a4:9c:9b:47:17:c0:7e:ae:60:29:f3:
        18:9b:af:ce:39:89:7e:bb:c7:99:70:fe:30:a0:e1:21:e1:99:
        4b:5d:91:be:a0:d6:4f:56:f4:56:46:c4:9b:e2:2f:f6:68:82:
        b1:f9:76:fb:38:24:08:98:04:62:fd:2c:8c:16:40:69:c6:f0:
        41:84:cf:b5:4f:e0:b1:46:84:80:33:cc:79:c8:77:cf:9a:2e:
        7d:21:39:64:2f:2c:26:a1:59:cb:93:39:0a:e9:81:bb:cb:d4:
        f6:38:1c:d4:db:fb:5d:62:56:aa:b8:d0:41:1b:7f:e7:b9:a1:
        d5:20:f0:09:68:a8:cb:f5:89:5f:15:ec:df:9c:12:04:56:75:
        32:9a:35:00:6a:be:e3:b6:74:8d:4f:5a:7a:d9:81:49:85:ab:
```

Secure Connection Failed

An error occurred during a connection to 10.99.0.7. Peer's Certificate has been revoked.

Error code: SEC_ERROR_REVOKED_CERTIFICATE

- The page you are trying to view cannot be shown because the authenticity of the received data could not be verified.
- Please contact the website owners to inform them of this problem.

Image 6

Certificate revoked on the Root CA and webserver is no longer accessible because the cert is revoked. OCSP responder reports it correctly.

3.2 - Authenticate to your webserver using a client certificate

Before we continue, we need to issue a new certificate for the webserver, so we going back to commands from `## 2.1 - Getting a certificate for your webserver`:
In webserver:

```
openssl req -new -addext 'subjectAltName = IP:10.99.0.7' -nodes -keyout
tls/webserver_new.key.pem -out webserver_new.csr.pem
scp webserver_new.csr.pem kali@10.99.0.15:/home/kali/rootca/requests
```

In rootca:

```
openssl ca -config openssl.cnf -extensions server_cert -in
requests/webserver_new.csr.pem -out certs/webserver_new.crt.pem
scp certs/webserver_new.crt.pem kali@10.99.0.7:/home/kali/webserver/tls
```

Back to webserver, as we changed the name of the cert, we need to edit `index.js`:

```
const tlsServerKey = fs.readFileSync('./tls/webserver_new.key.pem');
const tlsServerCrt = fs.readFileSync('./tls/webserver_new.crt.pem');
```

And now we are ready to communicate with our webserver with a new and valid certificate.

> Note: It is, in theory, possible to re-revoke the revoked cert but it is never safe to do so.

3.2.1 - Prepare your webserver

There are various ways to achieve so, one would be:

```
const clientAuth = () => (req, res, next) => {
  if (!req.client.authorized) {
    return res.status(401).send('Invalid cient certificate authentication.')
  }
  return next();
}

app.use(logger('dev'), clientAuth());
```

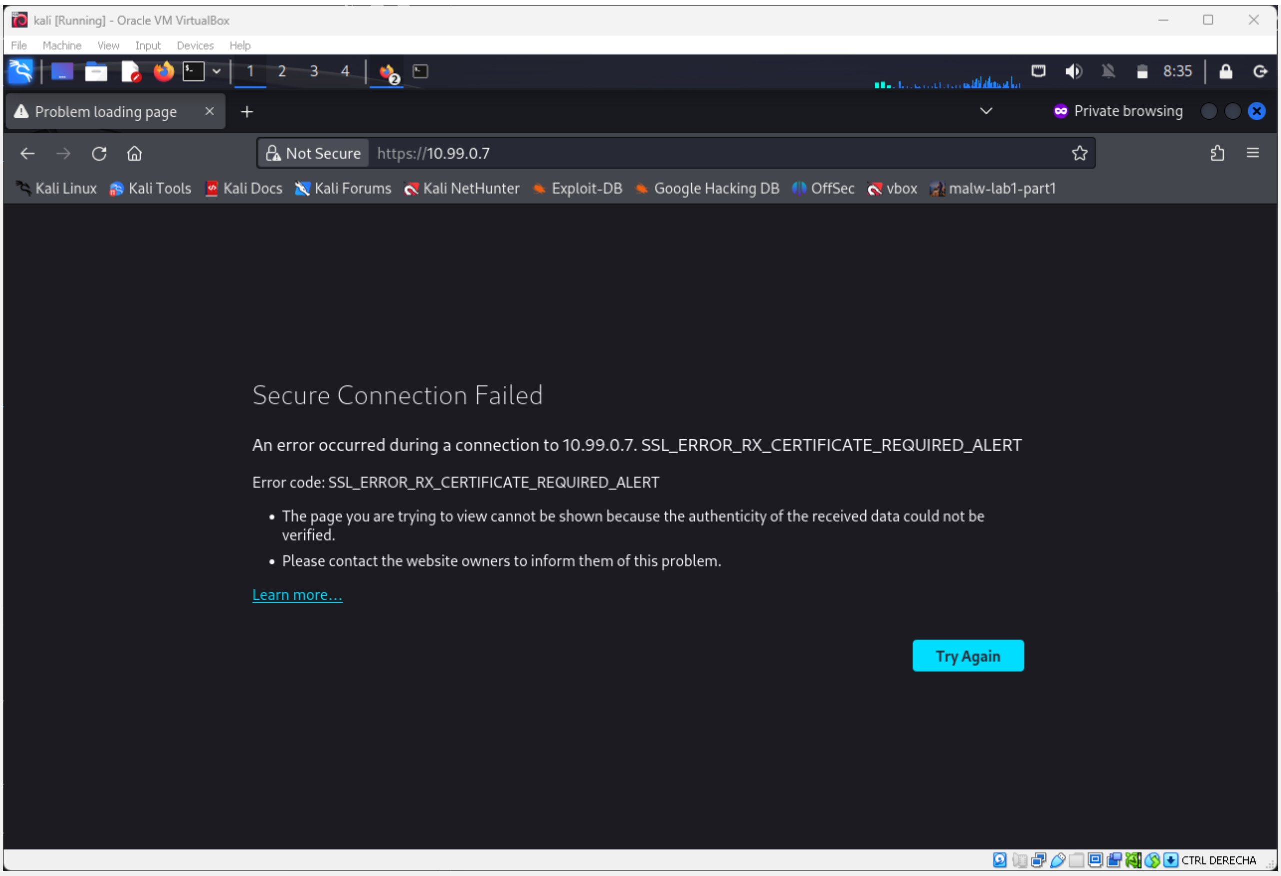
But this is a middleware approach. It could potentially introduce complexities or differences in the way the browser and server handle client certificate authentication.

So, instead, we using a method that aligns with the standard SSL/TLS handshake process. The final webserver code would be:

```
'use strict';

const express = require('express');
const logger = require('morgan');
const https = require('https');
const fs = require('fs');
const tlsServerKey = fs.readFileSync('./tls/webserver_new.key.pem');
const tlsServerCrt = fs.readFileSync('./tls/webserver_new.crt.pem');
const caCert = fs.readFileSync('./tls/cacert.pem');

const app = express();

app.use(logger('dev'));

app.get('/', (request, response) => {
    response.send('<h1>Hello Express</h1>');
});

const httpsOptions = {
    key: tlsServerKey,
    cert: tlsServerCrt,
    ca: caCert, // Provide the CA cert
    requestCert: true, // Request client cert
    rejectUnauthorized: true // Reject connections without a valid client certificate
};
const server = https.createServer(httpsOptions, app);

/**
 * Listen on provided port, on all network interfaces.
 */
server.listen(443);
server.on('listening', onListening);

/**
 * Event listener for HTTP server "listening" event.
 */
function onListening() {
    const addr = server.address();
    const bind = typeof addr === 'string'
        ? 'pipe ' + addr
        : 'port ' + addr.port;
    console.log('Listening on ' + bind);
}
```

In this case, we are never entering the website because the headers kick us out, which is more secure.

> Before we continue, let's check what happens if we try to access the website without the client certificate:


kali [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Problem loading page
Private browsing
Not Secure https://10.99.0.7
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec vbox malw-lab1-part1
Try Again
CTRL DERECHA

Secure Connection Failed

An error occurred during a connection to 10.99.0.7. SSL_ERROR_RX_CERTIFICATE_REQUIRED_ALERT

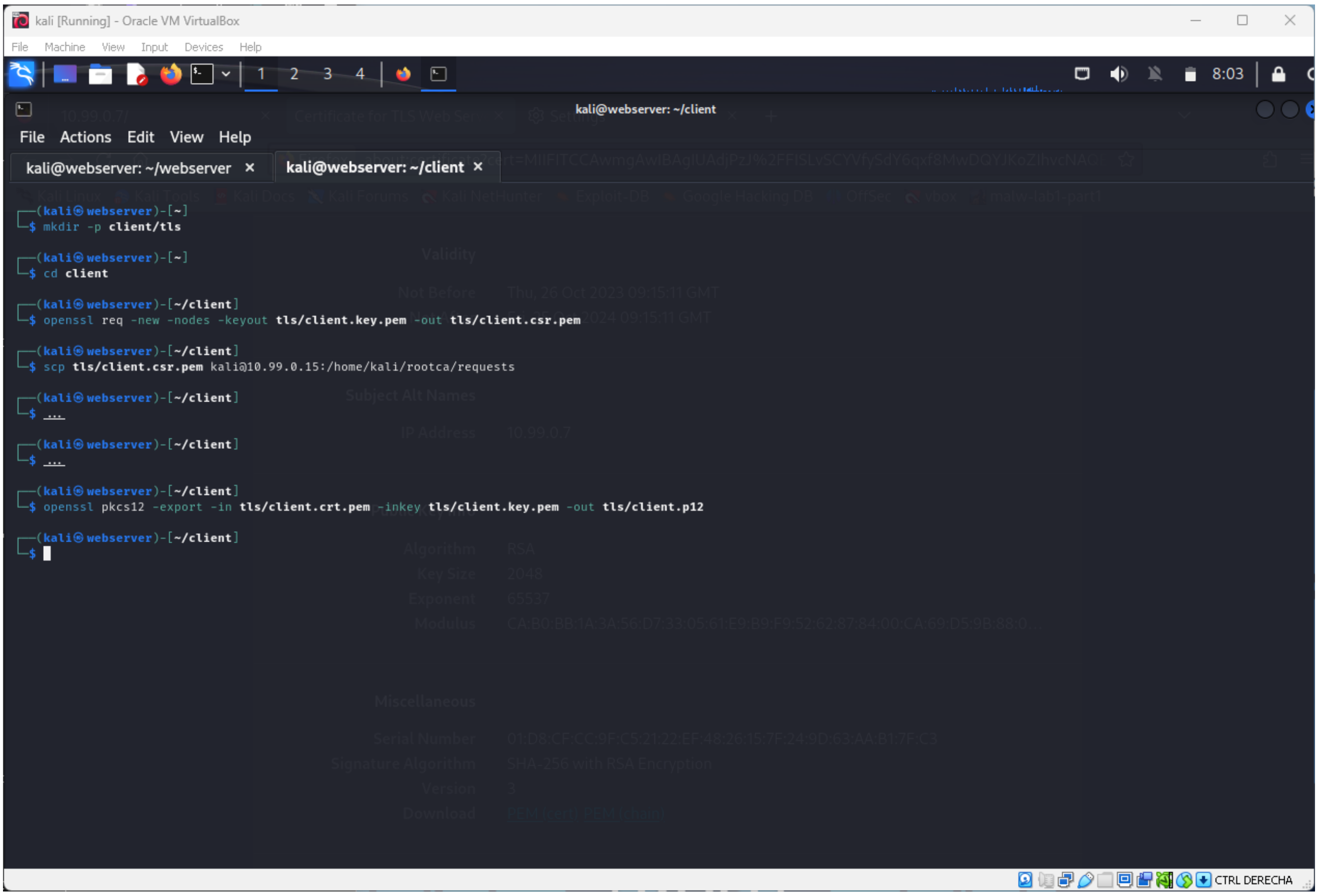
Error code: SSL_ERROR_RX_CERTIFICATE_REQUIRED_ALERT

- The page you are trying to view cannot be shown because the authenticity of the received data could not be verified.
- Please contact the website owners to inform them of this problem.

Learn more…

> As expected, we are not allowed to access the website.

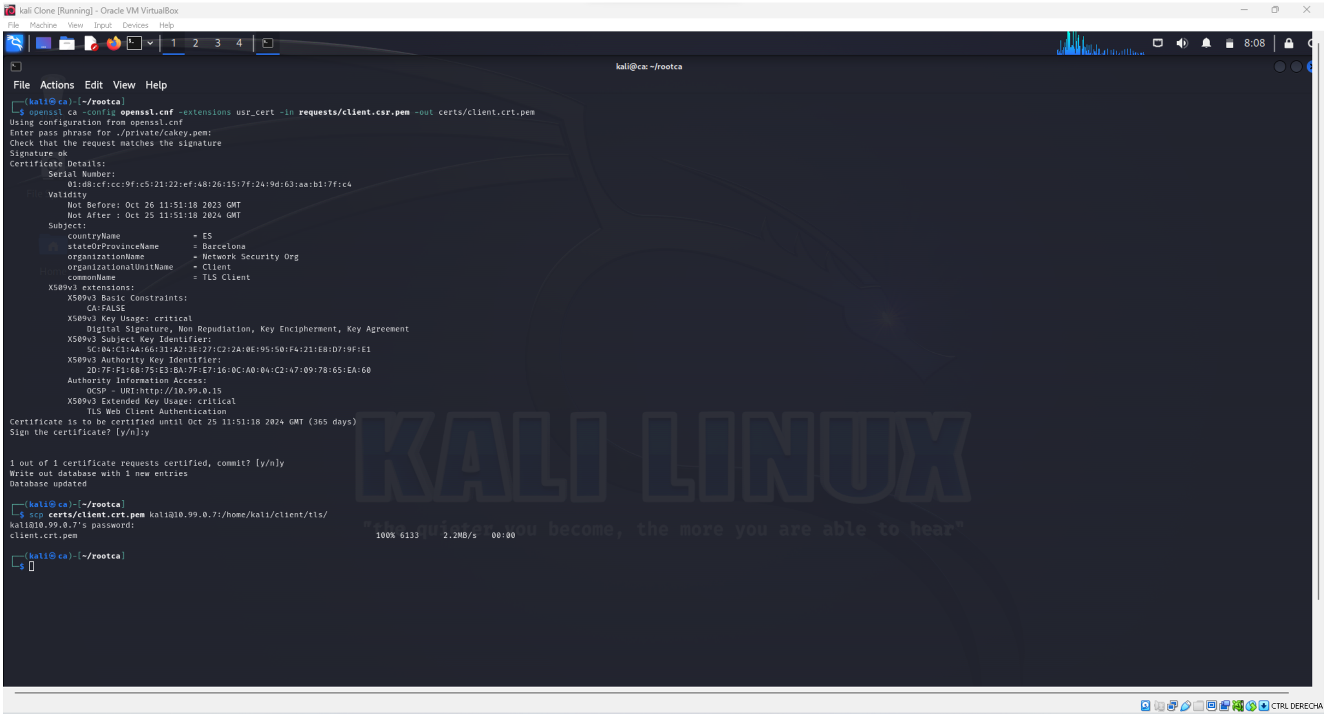
3.2.2 - Prepare your client


kali [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
kali@webserver: ~/client
File Actions Edit View Help
kali@webserver: ~/webserver
kali@webserver: ~/client
CTRL DERECHA

```
┌──(kali㉿webserver)-[~]
└─$ mkdir -p client/tls

┌──(kali㉿webserver)-[~]
└─$ cd client

┌──(kali㉿webserver)-[~/client]
└─$ openssl req -new -nodes -keyout tls/client.key.pem -out tls/client.csr.pem

┌──(kali㉿webserver)-[~/client]
└─$ scp tls/client.csr.pem kali@10.99.0.15:/home/kali/rootca/requests

┌──(kali㉿webserver)-[~/client]
└─$ ...

┌──(kali㉿webserver)-[~/client]
└─$ ...

┌──(kali㉿webserver)-[~/client]
└─$ openssl pkcs12 -export -in tls/client.crt.pem -inkey tls/client.key.pem -out tls/client.p12

┌──(kali㉿webserver)-[~/client]
└─$ █
```

Image 8

We create a key and cert for the client, and send the cert to the root CA for it to sign it.

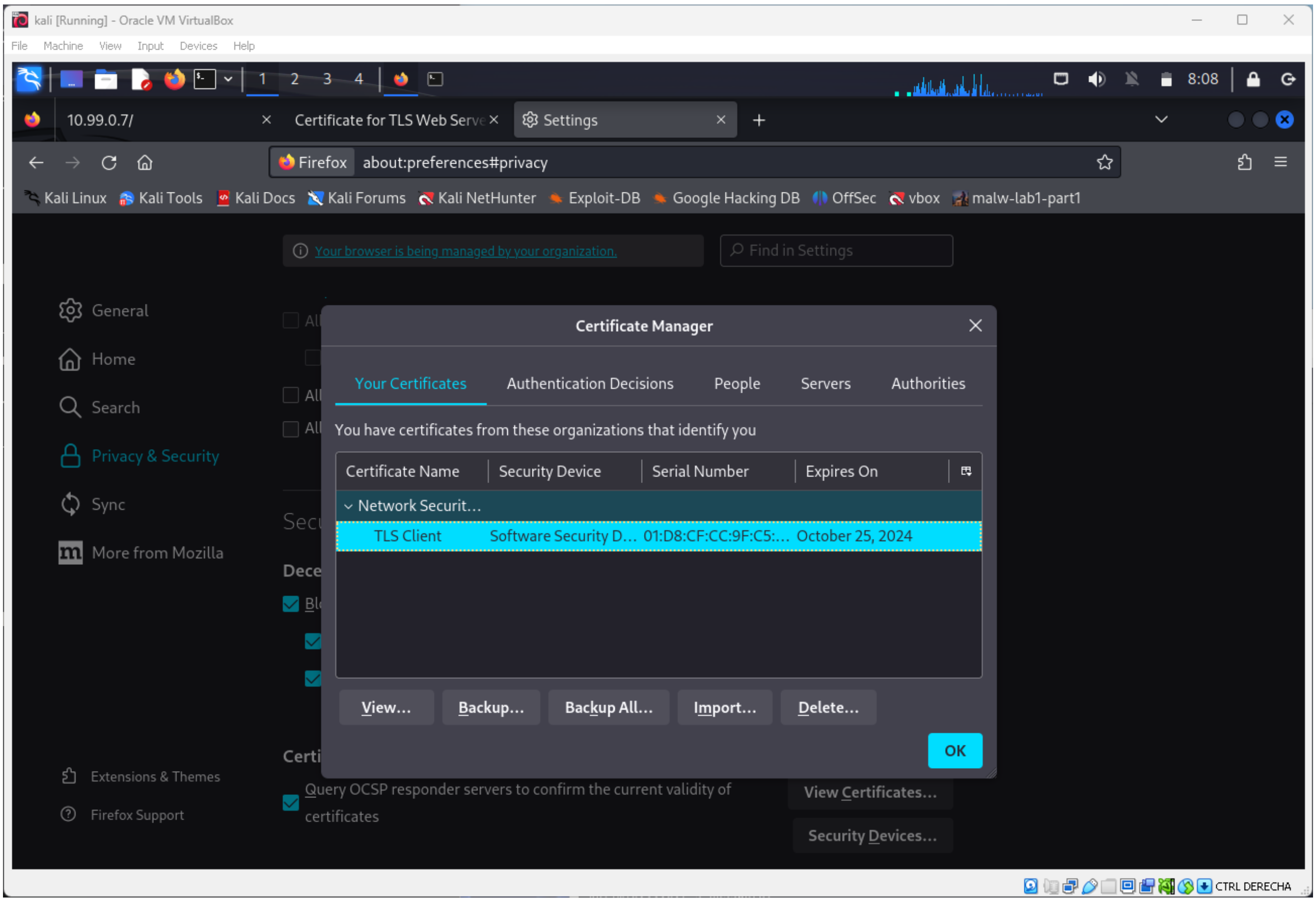


```
┌──(kali㉿ca)-[~/rootca]
└─$ openssl ca -config openssl.cnf -extensions usr_cert -in requests/client.csr.pem -out certs/client.crt.pem
Using configuration from openssl.cnf
Enter pass phrase for ./private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number:
            01:d8:cf:cc:9f:c5:21:22:ef:48:26:15:7f:24:9d:63:aa:b1:7f:c4
        Validity
            Not Before: Oct 26 11:51:18 2023 GMT
            Not After : Oct 25 11:51:18 2024 GMT
        Subject:
            countryName               = ES
            stateOrProvinceName       = Barcelona
            organizationName          = Network Security Org
            organizationalUnitName    = Client
            commonName                = TLS Client
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Key Usage: critical
                Digital Signature, Non Repudiation, Key Encipherment, Key Agreement
            X509v3 Subject Key Identifier:
                5C:04:C1:4A:66:31:A2:3E:27:C2:2A:0E:95:50:F4:21:E8:D7:9F:E1
            X509v3 Authority Key Identifier:
                2D:7F:F1:68:75:E3:BA:7F:E7:16:0C:A0:04:C2:47:09:78:65:EA:60
            Authority Information Access:
                OCSP - URI:http://10.99.0.15
            X509v3 Extended Key Usage: critical
                TLS Web Client Authentication
Certificate is to be certified until Oct 25 11:51:18 2024 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Database updated

┌──(kali㉿ca)-[~/rootca]
└─$ scp certs/client.crt.pem kali@10.99.0.7:/home/kali/client/tls/
kali@10.99.0.7's password:
client.crt.pem                                                  100% 6133     2.2MB/s   00:00

┌──(kali㉿ca)-[~/rootca]
└─$ 
```

Image 9

Root CA signs it and return it to the client.

Back to `Image 7`, we generate a `PKCS #12 file` with the key and the signed cert.



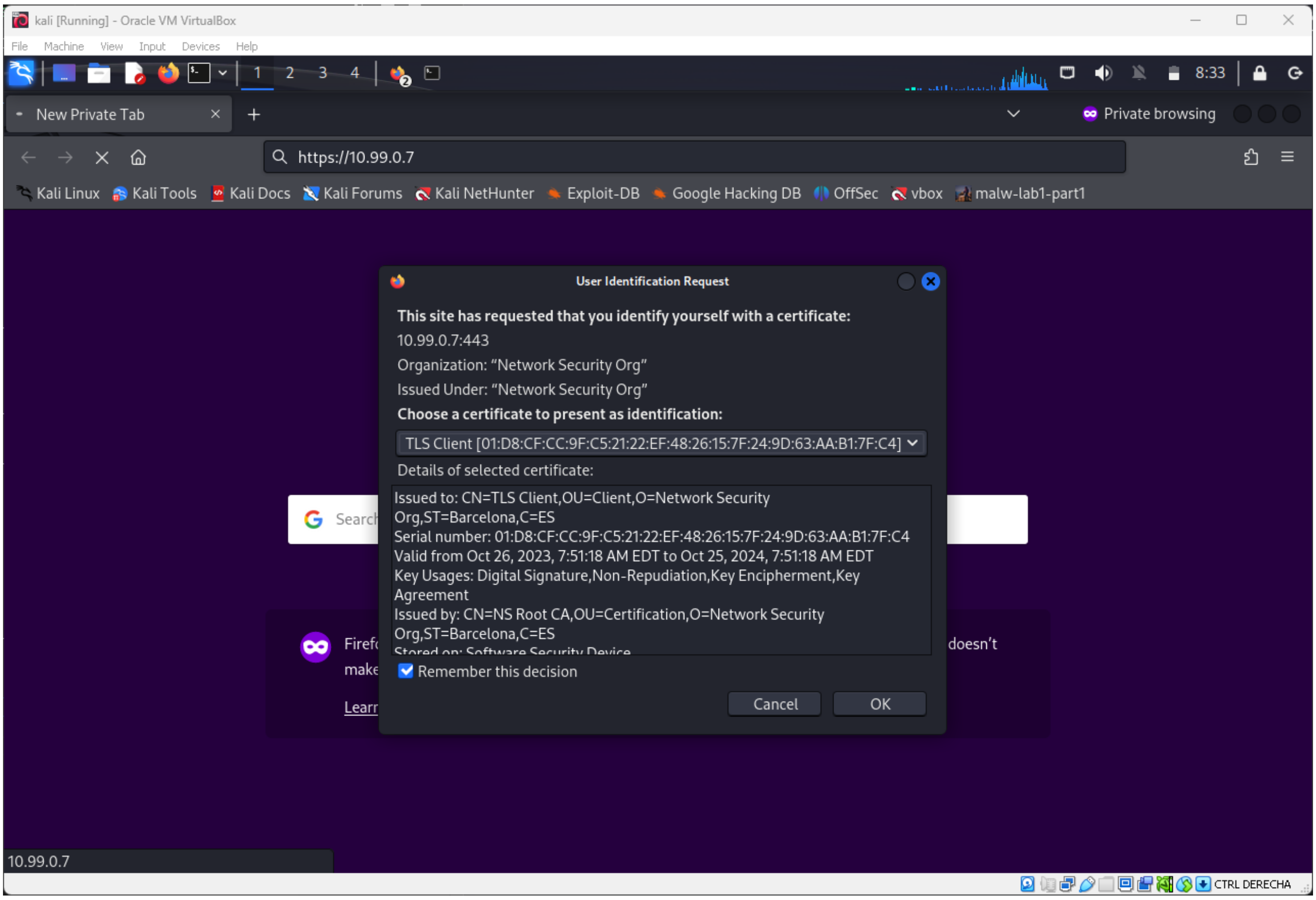
You have certificates from these organizations that identify you

Certificate Name	Security Device	Serial Number	Expires On
⌄ Network Securit...			
TLS Client	Software Security D...	01:D8:CF:CC:9F:C5:...	October 25, 2024

☑ Query OCSP responder servers to confirm the current validity of certificates

Image 10

We import that `.p12` file to firefox.



User Identification Request

This site has requested that you identify yourself with a certificate:

10.99.0.7:443

Organization: "Network Security Org"

Issued Under: "Network Security Org"

Choose a certificate to present as identification:

TLS Client [01:D8:CF:CC:9F:C5:21:22:EF:48:26:15:7F:24:9D:63:AA:B1:7F:C4]

Details of selected certificate:

Issued to: CN=TLS Client,OU=Client,O=Network Security Org,ST=Barcelona,C=ES
Serial number: 01:D8:CF:CC:9F:C5:21:22:EF:48:26:15:7F:24:9D:63:AA:B1:7F:C4
Valid from Oct 26, 2023, 7:51:18 AM EDT to Oct 25, 2024, 7:51:18 AM EDT
Key Usages: Digital Signature,Non-Repudiation,Key Encipherment,Key Agreement
Issued by: CN=NS Root CA,OU=Certification,O=Network Security Org,ST=Barcelona,C=ES
Stored on: Software Security Device

☑ Remember this decision

Cancel OK

Image 11

Once we access the website, firefox asks us if that cert is the one to be used and... *viola* we are inside!