Projecte ASIX 2k22

Escola Del Treball

2HISX 2021-2022

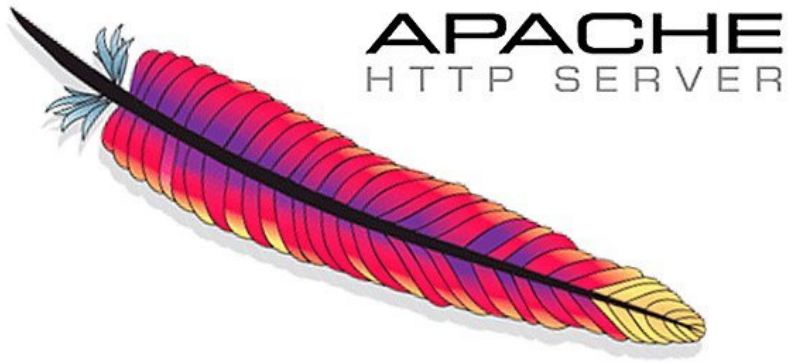
Aaron Andal & Cristian Condolo

CryptoSEC: "*Careful where you step in*"

Index

Apache2

Apache és un servidor web de codi obert, multiplataforma i gratuït. Aquest web server és un dels més utilitzats al món, actualment el 43% dels llocs web funcionen amb ell.



Apache té una estructura basada en mòduls, que permet activar i desactivar funcionalitats addicionals, per exemple, mòduls de seguretat com mod_security, mòduls de memòria cau com Varnish , o de personalització de capçaleres com mod_headers.

També permet ajustar els paràmetres de PHP del teu hosting de forma personalitzada mitjançant el fitxer .htaccess .

Avantatges

Els principals avantatges d'usar aquest servei web són els següents:

- De codi *obert* i **gratuït**, amb una gran comunitat dusuaris.
- Pegats de seguretat regulars i actualitzats amb freqüència.
- Estructura basada en **mòduls**.
- Multiplataforma. Està disponible a servidors **Windows** i **Linux**.
- Personalització mitjançant **.htaccess** independent a cada hosting.
- Compatible amb els principals **CMS** i **botigues online** i plataformes **e-learning**

Desavantatges

- Presenta problemes d'estabilitat per sobre de les 10.000 connexions
- Un ús abusiu de mòduls poden generar bretxes de seguretat.

Instal · lació Apache2 per a SOA CryptoSEC

1. Actualitzar el **repositori** i instal · lar **apache2**.

```
sudo apt update
```

```
sudo apt install apache2
```

2. Readjustar el firewall perque permeti Apache2. Apache es registra amb UFW per proporcionar alguns perfils d'aplicació que es poden utilitzar per habilitar o deshabilitar l'accés a Apache a través del *firewall*.

Llistem els perfils de **ufw**.

```
sudo ufw app list
```

Com ho indica el resultat, hi ha tres perfils disponibles per a Apache:

- **Apache**: aquest perfil obre només el port 80 (tràfic web normal no xifrat)

- **Apache Full**: aquest perfil obre el port 80 (tràfic web normal no xifrat) i el port 443 (tràfic TLS/SSL xifrat)
- **Apache Secure**: aquest perfil obre només el port 443 (tràfic TLS/SSL xifrat)

```
Output
Available applications:
  Apache
  Apache Full
  Apache Secure
  OpenSSH
```

El resultat proporcionarà una llista del trànsit de HTTP que es permet:

```
sudo ufw allow 'Apache'
```

```
sudo ufw status
```

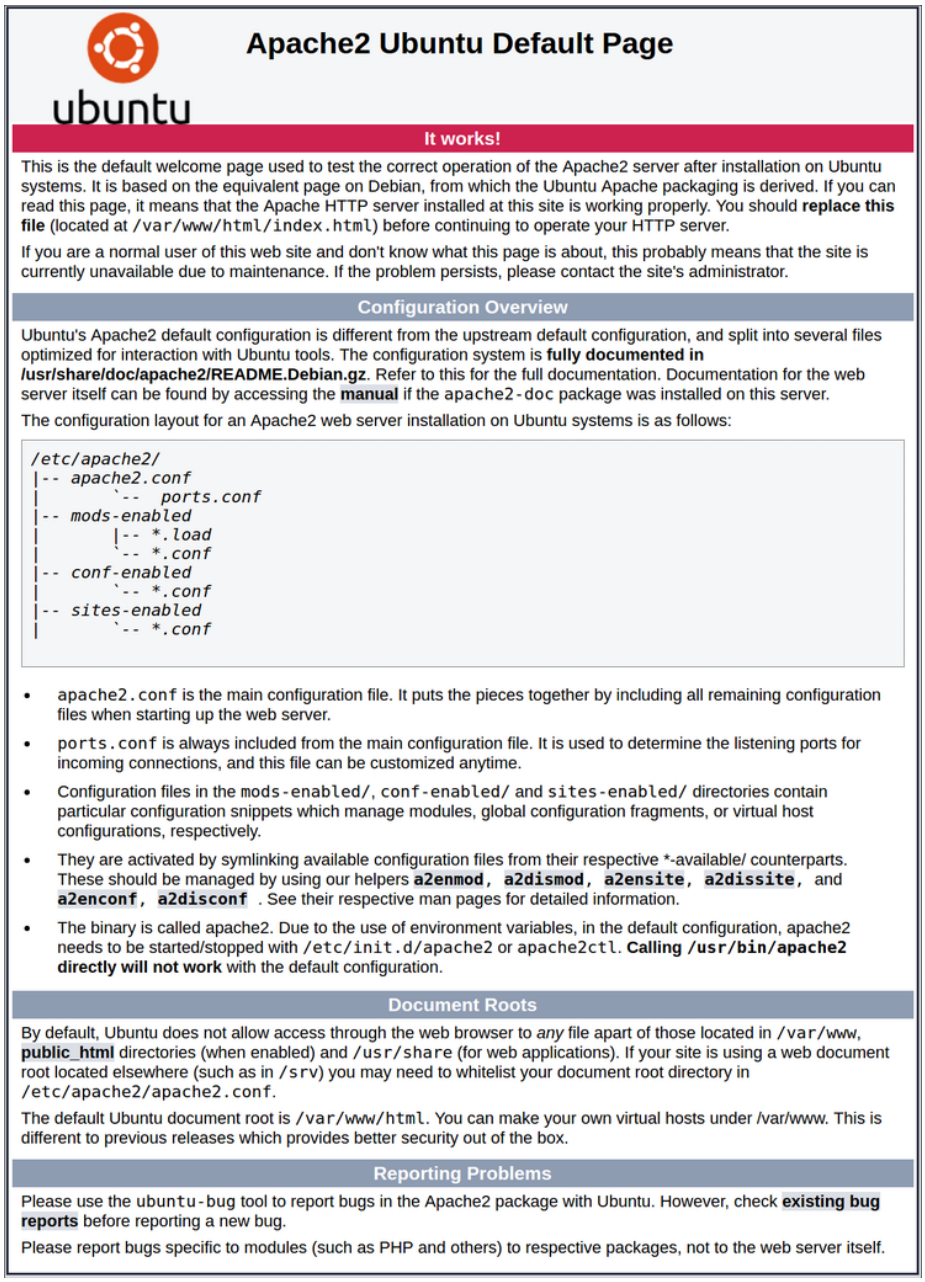
```
Output
Status: active

To                         Action      From
--                         ------      ----
OpenSSH                    ALLOW       Anywhere
Apache                     ALLOW       Anywhere
OpenSSH (v6)               ALLOW       Anywhere (v6)
Apache (v6)                ALLOW       Anywhere (v6)
```

3. Comprovar el servidor web.

```
sudo systemctl status apache2
```

```
Output
 apache2.service - The Apache HTTP Server
    Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
    Active: active (running) since Thu 2020-04-23 22:36:30 UTC; 20h ago
      Docs: https://httpd.apache.org/docs/2.4/
  Main PID: 29435 (apache2)
     Tasks: 55 (limit: 1137)
    Memory: 8.0M
    CGroup: /system.slice/apache2.service
            29435 /usr/sbin/apache2 -k start
            29437 /usr/sbin/apache2 -k start
            29438 /usr/sbin/apache2 -k start
```

4. Obrir un navegador i posar la IP del SOA 10.200.243.164. Per HTTP. Més tard configurarem HTTPS.



Apache2 Ubuntu Default Page

ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers **`a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`,** and **`a2enconf`, `a2disconf`** . See their respective man pages for detailed information.
- The binary is called apache2. Due to the use of environment variables, in the default configuration, apache2 needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

Document Roots

By default, Ubuntu does not allow access through the web browser to *any* file apart of those located in `/var/www`, **public_html** directories (when enabled) and `/usr/share` (for web applications). If your site is using a web document root located elsewhere (such as in `/srv`) you may need to whitelist your document root directory in `/etc/apache2/apache2.conf`.

The default Ubuntu document root is `/var/www/html`. You can make your own virtual hosts under /var/www. This is different to previous releases which provides better security out of the box.

Reporting Problems

Please use the `ubuntu-bug` tool to report bugs in the Apache2 package with Ubuntu. However, check **existing bug reports** before reporting a new bug.

Please report bugs specific to modules (such as PHP and others) to respective packages, not to the web server itself.

5. Administrar Apache2:

Per aturar el vostre servidor web, escriviu el següent:

```
sudo systemctl stop apache2
```

Per iniciar el servidor web quan no estigui actiu, escriviu el següent:

```
sudo systemctl start apache2
```

Per aturar i després iniciar el servei de nou, escriviu el següent:

```
sudo systemctl restart apache2
```

Si només feu canvis de configuració, Apache sovint es pot recarregar sense tancar connexions. Per fer-ho, utilitzeu aquesta ordre:

```
sudo systemctl reload apache2
```

Per defecte, Apache està configurat per iniciar automàticament quan el servidor ho fa. Si no és el que voleu, deshabiliteu aquest comportament escrivint el següent:

```
sudo systemctl disable apache2
```

Per tornar a habilitar el servei de manera que es carregui a l'inici, escriviu el següent:

```
sudo systemctl enable apache2
```

Configurar hosts virtuals

En utilitzar el servidor web **Apache**, podeu utilitzar hosts virtuals (similars a blocs de servidor de Nginx) per encapsular detalls de configuració i allotjar més d'un domini des d'un únic servidor.

El nostre domini es **cryptosec.net**. És a dir quan l'usuari posi **cryptosec.net** al navegador anirà a parar a la nostra pàgina web que està allotjada en **/var/www/html/cryptosec/**.

1. Crear el directori **cryptosec** a **/var/www/html**

```
sudo mkdir /var/www/html/cryptosec
```

2. Cambiem el owner.

```
sudo chown -R $USER:$USER /var/www/html/cryptosec
```

3. Els permisos dels roots web haurien de ser correctes si no vau modificar el valor umask, que estableix permisos de fitxers predeterminats.

```
sudo chmod -R 755 /var/www/html/cryptosec
```

4. A continuació, creeu una pàgina d'exemple **index.html** utilitzant vim o el vostre editor preferit:

```
sudo vim /var/www/html/cryptosec/index.html
```

5. El contingut tindrà això.

```
<html>
    <head>
        <title>Welcome to CRYPTOSEC.NET!</title>
    <style>
......
......
</html>
```

SEE FULL CONTENT –> ./index.html

6. Crear un fitxer de configuració per a la zona: **cryptosec.net.conf** que estarà a **/etc/apache2/sites_available**.

sudo nano /etc/apache2/sites-available/your_domain.conf

```
sudo nano /etc/apache2/sites-available/cryptosec.net.conf
```

Dins tindrà el VirtualHost, de moment el HTTP.

```
<VirtualHost *:80>
    ServerAdmin cryptosec@localhost
    ServerName cryptosec.net
    ServerAlias www.cryptosec.net
    DocumentRoot /var/www/cryptosec
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Tingueu en compte que canviem `DocumentRoot` pel nostre nou directori i `ServerAdmin` per un correu electrònic al qual pugui accedir l'administrador del lloc your_domain . També afegim dues directives: `ServerName`, que estableix el domini de base que hauria de coincidir per a aquesta definició de host virtual, i `ServerAlias`, que defineix més noms que haurien de coincidir com si fossin el nom de base.

7. Habilitat la zona del fitxer de configuració, per tenir validesa.

```
sudo a2ensite cryptosec.net
```

8. Provar que funciona, fem un config test d'Apache2.

```
sudo apache2ctl configtest
```

```
Output
Syntax OK
```

OPENSSL

Creació de claus i certificats

La TLS , o seguretat a la capa de transport, i la SSL , plataforma antecessora la sigla de la qual significa "capa de sockets segurs", són protocols web que s'utilitzen per embolicar el trànsit normal amb una cobertura protegida xifrada.

Mitjançant aquesta tecnologia, els servidors poden enviar trànsit de manera segura entre servidors i clients sense la possibilitat que els missatges siguin interceptats per tercers. El sistema de certificat també ajuda els usuaris a verificar la identitat dels llocs amb què estableixen connexió.

En aquesta guia, us mostrarem la manera de configurar un certificat SSL signat per Veritat Absoluta CA per a l'empresa CryptoSEC amb un servidor web d'Apache a Ubuntu 20.04.

CA VERITAT ABSOLUTA

1. Generem les claus per a la **CA Veritat Absoluta**.

```
openssl genrsa -out cakey.pem 1024
```

```
Generating RSA private key, 1024 bit long modulus (2 primes)
...........+++++
.................+++++
e is 65537 (0x010001)
```

2. Generem el certificat per de la **CA Veritat Absoluta**.

```
openssl req -new -x509 -nodes -sha1 -days 365 -key cakey.pem -out cacert.pem
```

```
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:CA
State or Province Name (full name) [Some-State]:Barcelona
Locality Name (eg, city) []:BCN
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Veritat Absoluta
Organizational Unit Name (eg, section) []:Veritat
Common Name (e.g. server FQDN or YOUR name) []:veritat.absoluta.net
Email Address []:veritat@absoluta.net
```

3. Generem el request per al certificat de CryptoSEC.net a partir de les claus de CA Veritat Absoluta.

```
openssl req -newkey rsa:2048 -nodes -sha256 -keyout cryptosec_key.pem -out cryptosec_req.pem
```

```
Generating a RSA private key
.......................+++++
........+++++
writing new private key to 'cryptosec_key.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:CA
State or Province Name (full name) [Some-State]:Barcelona
Locality Name (eg, city) []:BCN
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Cryptosec
Organizational Unit Name (eg, section) []:Cryptosec
Common Name (e.g. server FQDN or YOUR name) []:cryptosec.net
Email Address []:cryptosec@cryptosec.net

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:jupiter
An optional company name []:jupiter
```

4. Visualitzem:

```
cryptosec@SOACryptosec:~/CERTS$ ls -l
total 16
-rw-rw-r-- 1 cryptosec cryptosec 1143 May 13 20:50 cacert.pem
-rw------- 1 cryptosec cryptosec  887 May 13 20:49 cakey.pem
-rw------- 1 cryptosec cryptosec 1704 May 13 20:51 cryptosec_key.pem
-rw-rw-r-- 1 cryptosec cryptosec 1135 May 13 20:51 cryptosec_req.pem
```

5. Som CA, hem de firmar i generar el certificat per a CryptoSEC.

```
openssl x509 -CAkey cakey.pem -CA cacert.pem -req -in cryptosec_req.pem
 -days 3650 -CAcreateserial -out cryptosec_cert.pem
```

```
Signature ok
subject=C = CA, ST = Barcelona, L = BCN, O = Cryptosec, OU = Cryptosec, CN = cryptosec.net,
Getting CA Private Key
```

Habilitar SSL per HTTP

1. Habilitar **mod_ssl**, el mòdul SSL d'Apache i **mod_headers**, que necessiten algunes de les configuracions del nostre fragment SSL, amb l'ordre **a2enmod**:

```
sudo a2enmod ssl
sudo a2enmod headers
```

2. Modificar `/etc/apache2/sites-available/cryptosec.net.conf`.

```
sudo nano /etc/apache2/sites-available/cryptosec.net.conf
```

```
<VirtualHost *:443>
   ServerName cryptosec.net
   DocumentRoot /var/www/cryptosec

   SSLEngine on
   SSLCertificateFile /etc/ssl/certs/cryptosec_cert.pem
   SSLCertificateKeyFile /etc/ssl/private/cryptosec_key.pem
</VirtualHost>
```

3. Hem de copiar la clau i el certificat generat abans de Cryptosec.

```
cryptosec@SOACryptosec:~/CERTS$ sudo cp cryptosec_cert.pem /etc/ssl/certs/.
cryptosec@SOACryptosec:~/CERTS$ sudo cp cryptosec_key.pem /etc/ssl/private/.
```

- Verifiquem que estigui tot correcte:

```
sudo apache2ctl configtest
```

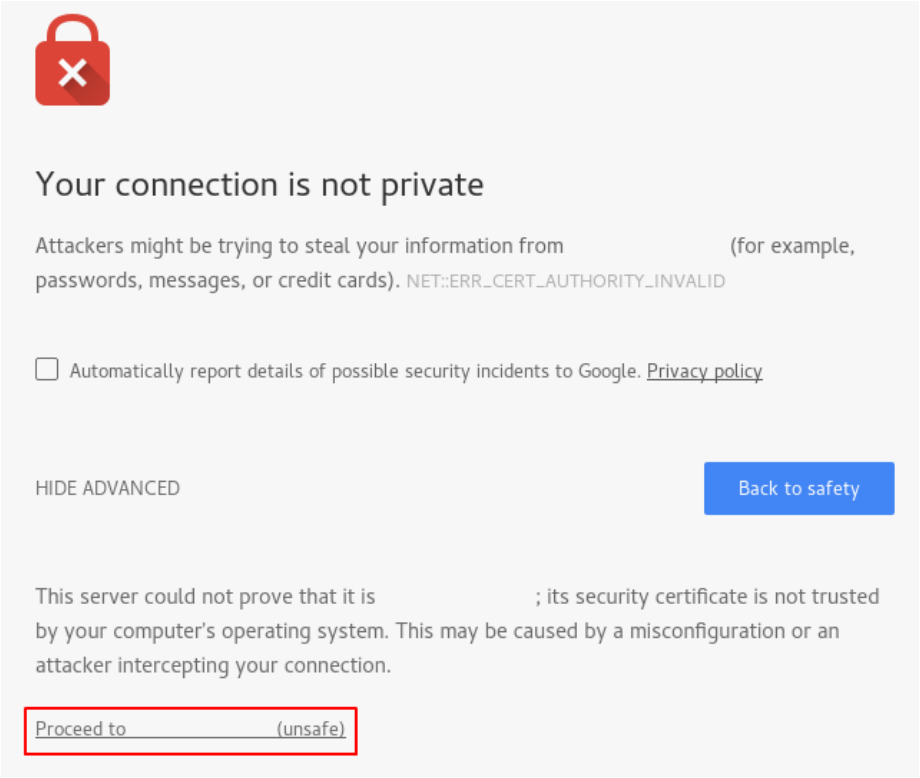
- Fem un reload per recarregar Apache2.

```
sudo systemctl reload apache2
```

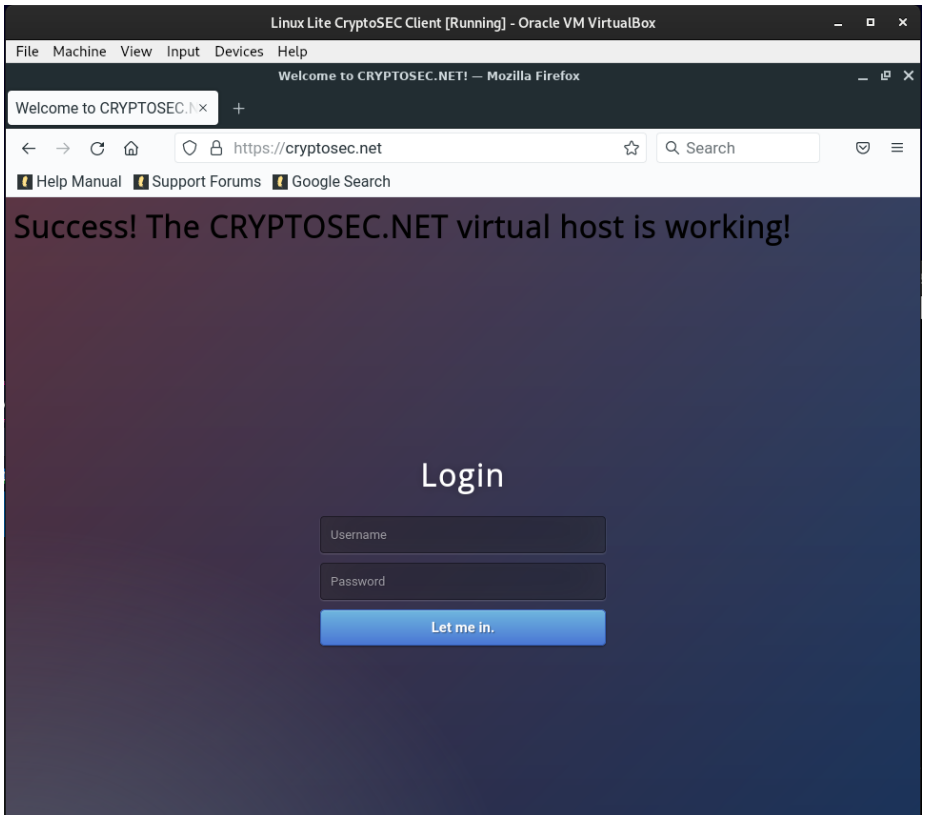
- Fem un reinici d'Apache2.

```
sudo systemctl restart apache2
```

4. Finalment provar-ho en un navegador, posant **https://cryptosec.net**, ens sortirà un missatge d'excepció, perquè el navegador no conèix o no té el certificat de la CA en el navegador, per tema de trust.



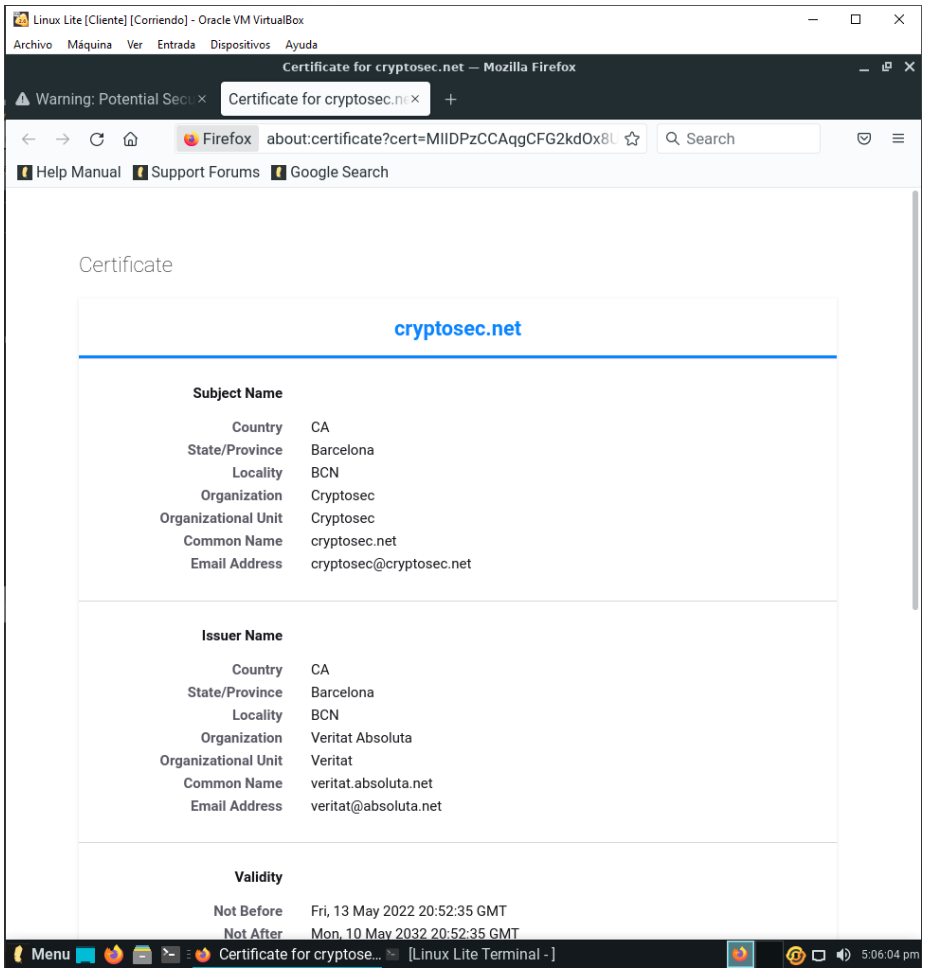
5. Per solucionar aquest problema, importem manualment el `cacert.pem` als navegadors.



6. Veure el contingut dels certificats:

```
openssl x509 --noout --text -in cryptosec_cert.pem
openssl x509 --noout --text -in cacert.pem
```

O des d'un navegador web Firefox:



Bibliografia

- https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-20-04-es - APACHE2
- https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-20-04-es - APACHE2