Prontuario per la realizzazione di un server WEB su https APACHE+MOD_SSL

Mirko Grava

mgrava@rhx.it

Questo documento descrive in modo pratico la realizzazione di un server WEB su https APACHE+MOD_SSL.

1. Introduzione

Per poter realizzare un server WEB con SSL si deve disporre di un'Authority che garantisca i nostri certificati. L'esempio tratterà la realizzazione di una Certification Authority (CA) Self-signed; quindi senza dover richiedere a terze parti di garantire la nostra identità.

Una volta realizzata la nostra CA potremmo tranquillamente realizzare tutti i certificati che vogliamo a condizione che la CA firmi i nostri certificati. L'idea di base è realizzare un server WEB con un numero n di VirtualHost e altrettanti certificati.

2. Software richiesto

Prima di iniziare assicuratevi di disporre del seguente software installato **apache, openssl, mod_ssl** e in modo facoltativo della documentazione **apache-manual**. I riferimenti specifici a mod_ssl si trovano in: `http://localhost/manual/mod/mod_ssl`.

3. Primi passi

Ogni server possiede almeno tre file:

- una chiave (`server.key`);
- un certificato da firmare (`server.csr`);

- un certificato firmato dalla CA: (`server.crt`).

La nostra CA Self-Signed possiede almeno due file:

- una chiave (`CA.key`);
- un certificato Self-Signed (`CA.crt`).

Solo due file in quanto si autocertifica e quindi si realizza solamente un file CA.crt senza passare per la realizzazione di un certificato da firmare da parte di una CA.

4. Si parte

Come utente **root** realizziamo la chiave primaria per la CA:

```
openssl genrsa -des3 -out CA.key 1024
```
si genera una chiave rsa cifrata des3 a 1024bit.

Realizziamo un certificato Self-Signed per la CA di durata 5 anni

```
openssl req -new -x509 -days 1825 -key CA.key -out CA.crt
```
si genera un certificato (crt) "autocertificato" di formato x509.

(IMPORTANTE segnatevi da qualche parte la passphrase).

Creiamo la directory di storage per la CA:

```
mkdir -p demoCA/{private,newcerts}
cp CA.crt demoCA/cacert.pem
cp CA.key demoCA/private/cakey.pem
echo 01 > demoCA/serial
```
(IMPORTANTE 01 in serial)
```
touch demoCA/index.txt
```

Tutto quello sopra ci serve per realizzare la struttura che il comando openssl cerca quando si firmano i certificati dei VirtualHost. Notate il fatto che la chiave e i certificati vengono rinominati da `CA.crt` a `cacert.pem` e da `CA.key` a `cakey.pem`

5. Il certificato per il VirtualHost

Per prima cosa la chiave:

```
openssl genrsa -des3 -out uno.key 1024
```
poi il certificato da firmare:
```
openssl req -new -key uno.key -out uno.csr
```

poi lo firmiamo tramite la CA che abbiamo creato:

```
openssl ca -in uno.csr -out uno.crt
```

Adesso abbiamo tutto quanto ci può servire per realizzare il primo VirtualHost.

```
cp CA.crt /etc/httpd/conf/ssl.crt/CA.crt
cp uno.crt /etc/httpd/conf/ssl.crt/uno.crt
cp uno.key /etc/httpd/conf/ssl.key/uno.key
```

(IMPORTANTE ricordarsi di fare chmod 600 di tutti i file indicati sopra). Questo provoca comunque la richiesta della passphrase tutte le volte che si riavvia il servizio httpd per evitare questo fare

```
cp uno.key uno.key.encrypted
openssl rsa -in uno.key.encrypted -out uno.key
cp uno.key /etc/httpd/conf/ssl.key/uno.key
```

6. configurazione apache httpd.conf

Cercare le righe

```
SSLEngine on
SSLCACertificateFile /etc/httpd/conf/ssl.crt/CA.crt
SSLCertificateFile /etc/httpd/conf/ssl.crt/uno.crt
SSLCertificateKeyFile /etc/httpd/conf/ssl.key/uno.key
```

(IMPORTANTE al fine di evitare errori nel parsing del file si ricorda che tutti i VirtualHost su porta 443 devono trovarsi tra la definizione: <IfDefine HAVE_SSL> e </IfDefine> relative alla parte di dichiarazione dei VirtualHost allegati anche i tre file di cui sopra + file .csr