

Linux Avanzato

Apache HTTP Server

Funzionalità

Configurazione

Virtual Host

SSL e https

Argomenti

- Apache httpd
- Directory importanti
- File di configurazione
- Permessi d'accesso
- Virtual Host
- Moduli ed estensioni
- SSL e https
- PHP

httpd.apache.org

The screenshot shows the Apache HTTP Server Project website. The browser's address bar displays 'httpd.apache.org'. The page header features the Apache feather logo and the text 'Apache HTTP SERVER PROJECT'. The main content area is divided into several sections:

- Essentials:** Includes links for 'About', 'License', 'FAQ', and 'Security Reports'.
- Download!:** Features a link 'From a Mirror'.
- Documentation:** Lists links for 'Version 2.4', 'Version 2.2', 'Trunk (dev)', and 'Wiki'.
- Get Support:** Includes a link for 'Support'.
- Get Involved:** Lists links for 'Mailing Lists', 'Bug Reports', and 'Developer Info'.
- Subprojects:** Lists links for 'Docs', 'Test', 'Flood', 'libapreq', 'Modules', 'mod_fcgid', and 'mod_ftp'.
- Miscellaneous:** Lists links for 'Contributors', 'Sponsors', and 'Sponsorship'.

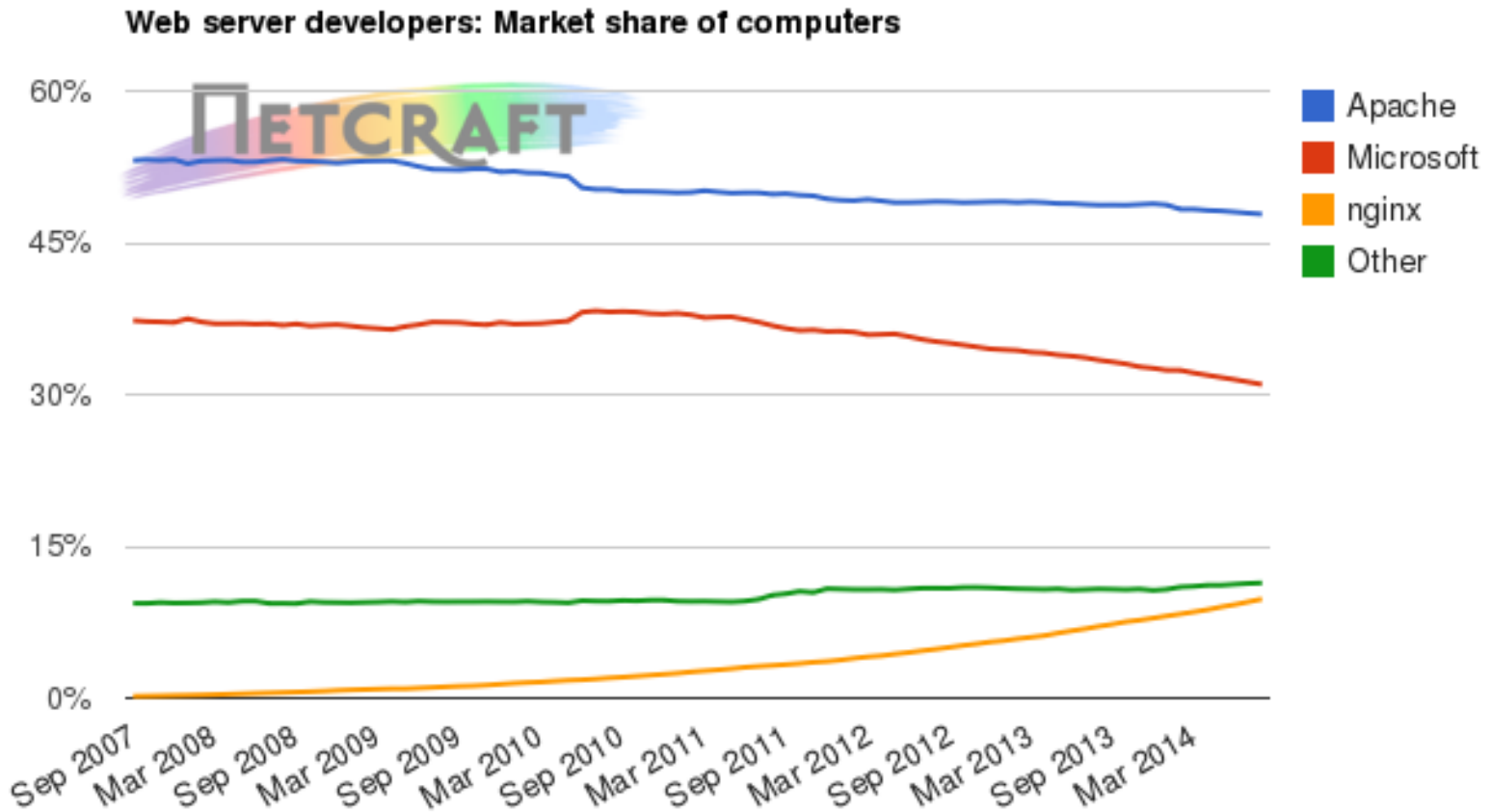
The main content area highlights two recent releases:

- Apache httpd 2.4.10 Released (2014-07-21):** Describes the release of version 2.4.10, noting it is the latest GA release and includes new features like Loadable MPMs and OSCP support.
- Apache httpd 2.2.29 Released (2014-09-03):** Describes the release of version 2.2.29, noting it is a security and bugfix release.

At the bottom of the page, there is a copyright notice: 'Copyright © 1997-2014 The Apache Software Foundation. Apache HTTP Server, Apache, and the Apache feather logo are trademarks of The Apache Software Foundation.'



Diffusione dei web server



Il lavoro del server http

- Rimani in attesa (LISTEN) di una richiesta http
- Analizza la richiesta
 - Estrae cookie e parametri
 - [statica] Restituisce il file
 - [dinamica] Attiva l'applicazione server
- Compone la risposta
- Invia la risposta
- Chiude la connessione e dimentica tutto

Orientarsi in Apache

- DocumentRoot
- Moduli
- Configurazione
- Logs
- Avvio processo
- Posizione dei file



```
sudo apt-get install apache2
```



DocumentRoot, URL, File

- DocumentRoot :
 - la cartella nella quale sono salvati i file statici distribuiti via http, o gli script dinamici attivati via http
 - Corrisponde alla “radice del sito web” : /
- Esempio:
<http://my.example.com/info/contact.html> viene mappato su {WebRoot}/info/contact.html
 - URL Mapping
 - URL Rewriting





Moduli

- La funzionalità http “base” è insita nel server (core)
- Molte funzionalità aggiuntive sono disponibili sotto forma di “moduli” (~100) che possono essere installati, configurati e attivati
- Ogni modulo si inserisce nella catena di elaborazione della richiesta
- Attivi su specifici tipi di file, o cartelle, o richieste, o fasi della risposta (hook)
- Esistono moduli di terze parti (es. mod_php).

Moduli installati

```

pcm2@pcm2:/usr/lib/apache2/modules$ ls
httpd.exp                mod_dav.so               mod_proxy_fcgi.so
libphp5.so              mod_dbd.so              mod_proxy_fdpass.so
mod_access_compat.so   mod_deflate.so         mod_proxy_ftp.so
mod_actions.so         mod_dialup.so          mod_proxy_html.so
mod_alias.so           mod_dir.so              mod_proxy_http.so
mod_allowmethods.so   mod_dumpio.so          mod_proxy_scgi.so
mod_asis.so           mod_echo.so            mod_proxy.so
mod_auth_basic.so     mod_env.so             mod_proxy_wstunnel.so
mod_auth_digest.so    mod_expires.so         mod_ratelimit.so
mod_auth_form.so      mod_ext_filter.so      mod_reflector.so
mod_authn_anon.so     mod_file_cache.so     mod_remoteip.so
mod_authn_core.so    mod_filter.so          mod_reqtimeout.so
mod_authn_dbd.so     mod_headers.so         mod_request.so
mod_authn_dbm.so     mod_heartbeat.so      mod_rewrite.so
mod_authn_file.so    mod_heartmonitor.so  mod_sed.so
mod_authn_socache.so  mod_include.so        mod_session_cookie.so
mod_authnz_ldap.so   mod_info.so           mod_session_crypto.so
mod_authz_core.so    mod_lbmethod_bybusyness.so
mod_authz_dbd.so     mod_lbmethod_byrequests.so
mod_authz_dbm.so     mod_lbmethod_bytraffic.so
mod_authz_groupfile.so
mod_authz_host.so    mod_lbmethod_heartbeat.so
mod_authz_owner.so  mod_ldap.so           mod_session_dbd.so
mod_authz_user.so   mod_log_debug.so     mod_session.so
mod_autoindex.so    mod_log_forensic.so  mod_setenvif.so
mod_buffer.so       mod_macro.so          mod_slotmem_plain.so
mod_cache_disk.so   mod_mime_magic.so    mod_slotmem_shm.so
mod_cache.so        mod_mime.so           mod_socache_dbm.so
mod_cache_socache.so
mod_cgid.so         mod_mpm_event.so     mod_socache_memcache.so
mod CGI.so          mod_mpm_prefork.so   mod_socache_shmcb.so
mod_charset_lite.so mod_mpm_worker.so    mod_speling.so
mod_data.so         mod_negotiation.so   mod_ssl.so
mod_dav_fs.so       mod_proxy_ajp.so     mod_status.so
mod_dav_lock.so    mod_proxy_balancer.so
mod_proxy_ajp.so    mod_proxy_connect.so mod_substitute.so
mod_proxy_balancer.so
mod_proxy_connect.so mod_proxy_express.so mod_suexec.so
mod_proxy_express.so
mod_proxy_fcgi.so   mod_proxy_ftp.so     mod_unique_id.so
mod_proxy_fdpass.so
mod_proxy_ftp.so   mod_proxy_html.so   mod_userdir.so
mod_proxy_html.so  mod_proxy_http.so   mod_usertrack.so
mod_proxy_http.so  mod_proxy_scgi.so   mod_vhost_alias.so
mod_proxy_scgi.so  mod_proxy.so        mod_xml2enc.so
mod_proxy_wstunnel.so
mod_ratelimit.so
mod_reflector.so
mod_remoteip.so
mod_reqtimeout.so
mod_request.so
mod_rewrite.so
mod_sed.so
mod_session_cookie.so
mod_session_crypto.so
mod_session_dbd.so
mod_session.so
mod_setenvif.so
mod_slotmem_plain.so
mod_slotmem_shm.so
mod_socache_dbm.so
mod_socache_memcache.so
mod_socache_shmcb.so
mod_speling.so
mod_ssl.so
mod_status.so
mod_substitute.so
mod_suexec.so
mod_unique_id.so
mod_userdir.so
mod_usertrack.so
mod_vhost_alias.so
mod_xml2enc.so
pcm2@pcm2:/usr/lib/apache2/modules$

```



Configurazione

- `/etc/apache2/apache2.conf`
 - File da esso inclusi
- Il file è diviso in **sezioni** `<sezione> ... </sezione>`
- Ogni sezione contiene una serie di **direttive**
Parametro = Valore
- Direttive interpretate sia dal server http, sia dai moduli di estensione caricati





Log

- /var/log/apache2/
- Access log
- Error log
- Rotazione dei log

```
pcm2@pcm2:/var/log/apache2$ ls -l
total 12
-rw-r----- 1 root adm 1348 Sep 20 15:46 access.log
-rw-r----- 1 root adm 5283 Sep 20 14:56 error.log
-rw-r----- 1 root adm  0 Aug 31 14:50 other_vhosts_access.log
pcm2@pcm2:/var/log/apache2$
```





Gestione dei processi httpd

```
$ sudo /etc/init.d/apache2 start  
$ sudo /etc/init.d/apache2 stop  
$ sudo /etc/init.d/apache2 restart  
$ sudo /etc/init.d/apache2 reload  
$ /etc/init.d/apache2 status
```

```
$ sudo service apache2 start  
$ sudo service apache2 stop  
$ sudo service apache2 restart  
$ sudo service apache2 reload  
$ service apache2 status
```





Posizione dei file

Variabile	Posizione
ServerRoot	<code>/etc/apache2</code>
Configurazione	<code>/etc/apache2</code>
Log	<code>/var/log/apache2</code>
Sito principale	<code>sites-enabled/000-default.conf</code>
DocumentRoot	<code>/var/www/html</code>
Moduli	<code>/usr/lib/apache2/modules</code>
Variabili d'ambiente	<code>/etc/apache2/envvars</code>
PID file	<code>/var/run/apache2/apache2.pid</code>





Posizione dei file

Variabile	Posizione
ServerRoot	/etc/apache2
Configurazione	/etc/apache2
Log	/var
Sito principale	site
DocumentRoot	/var
Moduli	/usr
Variabili d'ambiente	/etc
PID file	/var

```

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

```



1. *apache2.conf*: the main Apache2 configuration file. Contains settings that are *global* to Apache2.
2. *httpd.conf*: historically the main Apache2 configuration file, named after the *httpd* daemon. Now the file does not exist. In older versions of Ubuntu the file might be present, but empty, as all configuration options have been moved to the below referenced directories.
3. *conf-available*: this directory contains available configuration files. All files that were previously in `/etc/apache2/conf.d` should be moved to `/etc/apache2/conf-available`.
4. *conf-enabled*: holds *symlinks* to the files in `/etc/apache2/conf-available`. When a configuration file is symlinked, it will be enabled the next time *apache2* is restarted.
5. *envvars*: file where Apache2 *environment* variables are set.
6. *mods-available*: this directory contains configuration files to both load *modules* and configure them. Not all modules will have specific configuration files, however.
7. *mods-enabled*: holds *symlinks* to the files in `/etc/apache2/mods-available`. When a module configuration file is symlinked it will be enabled the next time *apache2* is restarted.
8. *ports.conf*: houses the directives that determine which TCP ports Apache2 is listening on.
9. *sites-available*: this directory has configuration files for Apache2 *Virtual Hosts*. Virtual Hosts allow Apache2 to be configured for multiple sites that have separate configurations.
10. *sites-enabled*: like *mods-enabled*, *sites-enabled* contains symlinks to the `/etc/apache2/sites-available` directory. Similarly when a configuration file in *sites-available* is symlinked, the site configured by it will be active once Apache2 is restarted.
11. *magic*: instructions for determining MIME type based on the first few bytes of a file.

<https://help.ubuntu.com/14.04/serverguide/httpd.html>

Attivazione/disattivazione funzionalità

- Ubuntu ha previsto diverse configurazioni
 - enabled: attualmente attive
 - Available: pronte per essere attivate (e configurate)
- Scripts: `a2{en|dis}{mod|conf|site}`

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


Creare pagine HTML

- Aggiungere file in DocumentRoot
 - /var/www/html
- Il processo “apache2” gira come utente “www-data”
 - File HTML leggibili da questo utente

Esercizio

- Aprire (e leggere) la pagina web di default installata da Ubuntu
- Creare un file prova .html nella DocumentRoot con del contenuto HTML qualsiasi
- Aprire via browser il file prova.html
- Verificare access log ed error log

Sintassi apache2.conf

- Il file di default contiene molti commenti esplicativi
- L'ordine delle direttive non è significativo
- Una direttiva per riga (“\” finale unisce righe successive)
- Direttive **case-insensitive**, molti argomenti **case-sensitive**
- Commenti: “#”

Direttive principali (1)

- **ServerRoot**
 - Directory principale per l'installazione
 - Molti path sono definiti come path relativi alla ServerRoot
- **DocumentRoot**
 - Directory di default corrispondente alla radice / del sito

Direttive principali (2)

- **Listen**

- Specifica su quali interfacce, indirizzi IP e porte il server rimane in ascolto
- Se specifico solo la porta, ascolta su tutti gli indirizzi IP della macchina

- **PidFile**

- File contenente il PID del processo apache principale
- I diversi thread del server sono processi figli di questo PID

Direttive principali (3)

- **DirectoryIndex**

- Nome del file da servire se la URI richiesta corrisponde al nome di una directory
- Solitamente: index.html

- **ErrorDocument**

- Specifica la pagina HTML generata quando si verifica un errore a livello HTTP
- Può essere un file di testo, un redirect locale o un redirect esterno

Direttive principali (4)

- **Alias**

- Associa una URI (o un prefisso di URI) ad una certa directory fisica
- `Alias /icons/ /usr/local/apache/icons/`

- **Redirect**

- Segnala al browser che una determinata URI è stata spostata (su questo o su un altro server) (HTTP 301)
- `Redirect /service`
`http://foo2.example.com/service`

Direttive principali (5)

- **Order/Allow/Deny**
 - Specificano i permessi e le restrizioni di accesso
 - Order: Deny, Allow *oppure* Order: Allow, Deny
 - Deny ed Allow specificano dei filtri di indirizzi IP, netmask
 - Deny from all
 - Allow from 192.168.54.*

Direttive principali (5)

- Order/Allow/Deny

- Specificare i permessi e le restrizioni di accesso

- Order Hosts whose names match, or end in, this string are allowed access. Only complete components are matched, so the above example will match `foo.example.org` but it will not match `fooexample.org`. This configuration will cause Apache httpd to perform a double DNS lookup on the client IP address, regardless of the setting of the `HOSTNAMELOOKUPS` directive. It will do a reverse DNS lookup on the IP address to find the associated hostname, and then do a forward lookup on the hostname to assure that it matches the original IP address. Only if the forward and reverse DNS are consistent and the hostname matches will access be allowed.

- **A full IP address** `Allow from 10.1.2.3`
`Allow from 192.168.1.104 192.168.1.205`
 An IP address of a host allowed access

- **A partial IP address** `Allow from 10.1`
`Allow from 10 172.20 192.168.2`
 The first 1 to 3 bytes of an IP address, for subnet restriction.

- **A network/netmask pair** `Allow from 10.1.0.0/255.255.0.0`
 A network a.b.c.d, and a netmask w.x.y.z. For more fine-grained subnet restriction.

- **A network/nnn CIDR specification** `Allow from 10.1.0.0/16`
 Similar to the previous case, except the netmask consists of nnn high-order 1 bits.

Direttive principali (6)

- **Options** – Configura le funzioni aggiuntive disponibili nelle directory
 - Indexes – Se manca DirectoryIndex, genera automaticamente l'elenco dei file presenti
 - AddIcon – Specifica immagini delle icone per i vari tipi di file
 - Multiviews – Attiva la negoziazione della lingua
 - ExecCGI – Permette l'esecuzione di script CGI
 - Includes – Attiva l'elaborazione di Server-Side includes o parsed HTML

Sezione <Directory>

- Specifica una serie di direttive che si applicano solamente ad un sotto-insieme dell'albero dei file
- Specifica il full path (fisico) a cui si applica

```
<Directory "/usr/local/httpd/htdocs">  
  Options Indexes FollowSymLinks  
</Directory>
```

```
<Directory /home/*/public_html>  
  Options Indexes  
</Directory>
```

Sezione <Files>

- Applica un insieme di directory a tutti i file il cui nome corrisponde

```
<Files private.html>  
    Require all denied  
</Files>
```

```
<Directory /var/web/dirl>  
    <Files private.html>  
        Require all denied  
    </Files>  
</Directory>
```

Sezione <Location>

- Applica le direttive quando la URI Richiesta (e non il path fisico) corrisponde

```
<Location /server-status>  
    SetHandler server-status  
</Location>
```

Il file `.htaccess`

- Oltre alle specifiche date nelle sezioni `<Directory>`, ogni directory può avere un file chiamato `.htaccess`
- Il `.htaccess` può contenere direttive aggiuntive, che si sovrappongono a quelle dichiarate nei file `.conf`
- Rallenta il server (deve essere letto per ogni richiesta)

Esercizio

- Creare un mini-sito
 - File HTML accessibili sotto la URL /sito (path fisico /var/www/sito)
 - File immagini accessibili sotto la URL /immsito (path fisico /var/www/immagini)

Virtual Hosts

- Più siti web, indipendenti, gestiti sullo stesso server
- Ogni virtual host ha una configurazione del tutto indipendente
 - ServerName, DocumentRoot, Aliases, log files, etc.
 - <http://httpd.apache.org/docs/current/vhosts/>

Virtual Hosts

- Apache supporta due tipi di virtual host
 - **Name-based** virtual host
 - Seleziona la configurazione virtuale sulla base del nome DNS utilizzato dal client
 - Permette molti virtual host sullo stesso IP address
 - Necessario agire sul DNS
 - **IP-based** virtual host
 - Seleziona la configurazione virtuale sulla base dell'IP address richiesto
 - Necessario che il server abbia più indirizzi IP, uno per ciascun sito

Name-Based Virtual Host

- **ServerName**
seleziona quale
virtual host attivare
- **ServerAlias** può
specificare nomi
aggiuntivi
- **DocumentRoot**
diversa

```
<VirtualHost *:80>  
    # This first-listed virtual host is  
    ServerName www.example.com  
    ServerAlias example.com  
    DocumentRoot /www/domain  
</VirtualHost>  
  
<VirtualHost *:80>  
    ServerName other.example.com  
    DocumentRoot /www/otherdomain  
</VirtualHost>
```

IP-Based Virtual Host

- Indirizzo IP diverso per I diversi host
- Direttive diverse per ciascun sito
- Saranno necessari comandi **Listen** per ciascun IP

```
<VirtualHost 172.20.30.40:80>
  ServerAdmin webmaster@www1.example.com
  DocumentRoot /www/vhosts/www1
  ServerName www1.example.com
  ErrorLog /www/logs/www1/error_log
  CustomLog /www/logs/www1/access_log combined
</VirtualHost>

<VirtualHost 172.20.30.50:80>
  ServerAdmin webmaster@www2.example.org
  DocumentRoot /www/vhosts/www2
  ServerName www2.example.org
  ErrorLog /www/logs/www2/error_log
  CustomLog /www/logs/www2/access_log combined
</VirtualHost>
```

Virtual Host su Ubuntu



- Prendere ad esempio il sito 000-default.conf
- Creare un nuovo file .conf per il nostro sito, nella directory -available
- Attivare il sito (a2ensite)

Esercizio

- Creare due virtual host IP-based
 - Indirizzo IP della rete Host Only
 - Porte di rete diverse (es. 80 ed 81)
 - Definire DocumentRoot distinte e verificare che funzioni
- Ripetere l'esercizio con virtual host name-based
 - Occorrerà modificare il file di **hosts** della macchina client e della macchina server

HTTPS e siti web sicuri

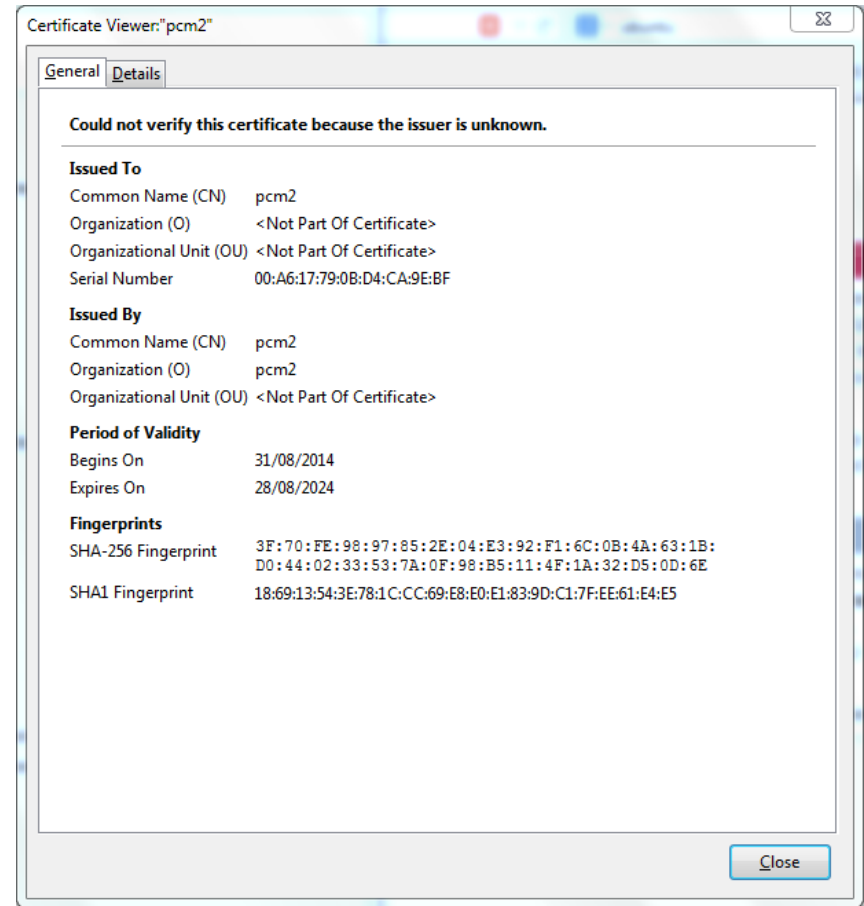
- La connessione HTTP è in chiaro
- E' possibile cifrare con SSL la connessione utilizzando il protocollo HTTPS (porta 443)
- Gestito dal modulo `mod_ssl` (nel package `apache2-common`)
- Necessario un «certificato server»

Attivazione https

- Attivare il modulo `mod_ssl`
 - `sudo a2enmod ssl`
- Verificare il sito di prova descritto in `/etc/apache2/sites-available/default-ssl.conf`
- Attivare il sito https di prova
 - `sudo a2ensite default-ssl`
- `sudo service apache2 restart`

Certificati?

- Il sito default-ssl.conf lavora con certificati self-signed
- Necessario sostituirli con certificati firmati da un'autorità riconosciuta



Direttive per https

- SSLEngine on
- SSLCertificateFile
 - Certificato in formato PEM
 - Può contenere la catena di certificazione
- SSLCertificateKeyFile
 - Contiene la chiave privata del certificato
 - Se la chiave privata è crittografata, sarà necessario inserire la passphrase all'avvio di apache

Esempio minimo

```
LoadModule ssl_module modules/mod_ssl.so

Listen 443
<VirtualHost *:443>
    ServerName www.example.com
    SSLEngine on
    SSLCertificateFile /path/to/www.example.com.cert
    SSLCertificateKeyFile /path/to/www.example.com.key
</VirtualHost>
```

Esercizio

- Attivare il server HTTPS di default
- Attivare il server HTTPS per almeno uno dei virtual host configurati prima.

Riferimenti

- <http://httpd.apache.org/>
- <https://help.ubuntu.com/14.04/serverguide/httpd.html>

These slides are licensed under a **Creative Commons**

**Attribution
Non Commercial
Share Alike
4.0 International**

To view a copy of this license, visit

<http://creativecommons.org/licenses/by-nc-sa/4.0/>

Versione in Italiano:

<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.it>

