20.1 WWW 的簡史、資源以及伺服器軟體

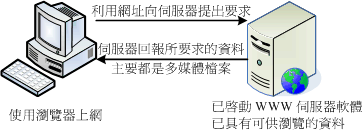
讓我們來瞭解瞭解什麼是WWW以及他所需要的伺服器軟體，還有一些瀏覽器相關的資訊吧！

20.1.1 WWW 的簡史、HTML 與標準制訂 (W3C)



WWW是World Wide Web的縮寫。WWW結合文字、圖形、影像以及聲音等多媒體，並透過超連結(Hyperlink)將資訊以 Internet傳遞到各處去。WWW server/client的相關性：

圖 20.1-1、WWW 伺服器與用戶端瀏覽器之間的連線相關性



從上面的圖示當中，我們大概可以得到一些觀念：

* WWW伺服器不但需要一個可讓用戶端瀏覽的平台，還需要提供用戶端一些資料才行！
* 伺服器所提供的最主要資料是超文件標籤語言(Hyper Text Markup Language, HTML)、多媒體檔案(圖片、影像、聲音、文字等，都屬於多媒體或稱為超媒體)。
* HTML只是一些純文字資料，透過所謂的標籤(<tag>)來規範所要顯示的資料格式；
* 在用戶端，透過瀏覽器的對HTML以及多媒體的解析，最後呈現在使用者的螢幕上。

HTML 的格式

伺服器端需要提供用戶端一些資料，而這些資料其實主要都以 HTML 的格式來呈現的。

|  |
| --- |
| <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="zh-TW" lang="zh-TW">  <head>  ....一些此頁面的資訊解釋的標頭資料，例如 title 與整體化設計等等....  </head>  <body style="margin:0; padding:0">  ....在瀏覽器顯示的畫面中，實際放置在瀏覽器上面的資料則寫於此....  </body>  </html> |

HTML是由<html> </html>所包含起來，其中又分為兩大區塊，一是標頭有關的<head> </head>區塊，包括該網頁所使用的編碼格式與抬頭等等。另一部份則是<body> </body>所含有的實際網頁內容資料。

WWW所用的協定及WWW伺服器簡史

為了讓HTTP得以順利應用，開發出伺服器HTTPd (HTTP daemon)。由於HTTPd沒妥善發展，改良過的軟體為Apache，取其『一個修改的伺服器(A patch server)』的雙關語！Apache在96年以後便成WWW伺服器上市佔率最高的軟體了。

20.1.2 WWW 伺服器與瀏覽器所提供的資源設定 (URL)



WWW伺服器的重點是提供一些資料，這些資料必需要是用戶端的瀏覽器可以支援顯示才行。這些資料大部分是檔案。必需要在伺服器端先將資料檔案寫好，並且放置在某個特殊的目錄底下，這個目錄就是整個網站的首頁了！一般來說，這個目錄可能在/var/www/html/或/srv/www/。瀏覽器如何取得這個目錄內的資料呢？在瀏覽器的『網址列』輸入所需要的網址才行。這個網址就對應到WWW伺服器的某個檔案檔名。現今的瀏覽器功能很多，不只可以連WWW，還可以連上類似FTP之類的網路通訊協定。所以你得要在網址列輸入正確的網址，這個網址包括這樣：

* <協定>://<主機位址或主機名稱>[:port]/<目錄資源>

網址列的意義

URL (Uniform Resource Locator)以斜線作為分段， 它可以這樣被解釋：

* 協定：瀏覽器比較常支援的協定有http, https, ftp, telnet等等。協定在告知瀏覽器『利用此一協定連接到伺服器端』。舉例來說，如果下達：<http://ftp.ksu.edu.tw>表示瀏覽器要連結到崑山科大的http(亦即 port 80)。如果是[ftp://ftp.ksu.edu.tw](ftp://ftp.ksu.edu.tw" \t "_blank)則代表連結到ftp(port 21)！使用的協定不同，回應的資料也不相同。不過，萬一對方伺服器的埠口啟動在非正規埠號，例如將http啟動在port 81時，就得這樣寫：http://hostname:81/ 。
* 主機位址或主機名稱：就是伺服器在網際網路所在的IP位置。如果是主機名稱，當然要透過名稱解析器！一般來說，雖然使用IP就能夠架設WWW網站，不過建議你還是申請一個好記又合法的主機名稱比較好！
* 目錄資源：在首頁目錄下的相對位置就是這個目錄資源。舉例來說，鳥哥的網站www資料放置在我主機的 /var/www/html/中，所以說：
  + http://linux.vbird.org --> /var/www/html/
  + http://linux.vbird.org/linux\_basic/index.php --> /var/www/html/linux\_basic/index.php

通常首頁目錄底下會有個特殊的檔案名稱，如index.html或index.???等。舉例來說，如果直接按下： http://linux.vbird.org會發現與http://linux.vbird.org/index.php一樣！因為WWW伺服器會主動的以該目錄下的『首頁』來顯示！伺服器會由於瀏覽器傳來的要求協定不同而給予不一樣的回應資料。

WWW server/client 間資料傳輸的方式

如果瀏覽器以http://hostname的型態來向伺服器要資料時，瀏覽器與伺服器端是如何傳遞資料的呢？有這幾種方法：

* GET: 瀏覽器直接向WWW伺服器要求網址列上面的資源，這也是最常見的。使用GET的方式可以直接在網址列輸入變數。舉例來說，網址：『<http://phorum.vbird.org/viewtopic.php?t=96>』，t就是變數，96就是這個變數的內容。如果你將問號後面的資料拿掉時，瞧瞧會出現什麼後果？這麼說，你可以明GET的處理了吧？
* POST: 用戶端向伺服器端提出的要求，只是這個要求裡含有較多的資料。舉例來說，討論區裡留言的選項，選擇留言會在瀏覽器冒出一個框框讓你填入資料！按下傳送後，框框內的資料會被瀏覽器包起來傳送至WWW伺服器。
* HEAD: 伺服器端回應給Client端的一些資料檔頭而已；
* OPTIONS: 伺服器端回應給Client端的一些允許的功能與方法；
* DELETE: 刪除某些資源的舉動。

常見的是GET！如有大量資料由用戶端上傳到WWW伺服器端時，才會使用到POST這個項目。

20.1.3 WWW伺服器的類型： 系統、平台、資料庫與程式 (LAMP)



以目前的網路世界來說，市佔率較高的WWW伺服器軟體應該是Apache與IIS，Apache是自由軟體，可在任何作業系統上安裝，IIS則是Windows家族開發出來的，僅能在Windows作業系統上面安裝與執行。

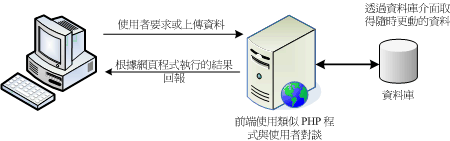
僅提供使用者瀏覽的單向靜態網頁

這種類型的網站大多是提供『單向靜態』的網頁，或許有提供一些動畫圖示，但基本上就僅止於此！因為單純是由伺服器單向提供資料給客戶端，Server不需與Client端互動，所以可到該網站上去瀏覽，但是無法進行資料的上傳！目前主要的免費虛擬主機大多是這種類型。只要依照HTML的語法寫好你的網頁，並上傳到該網站空間上，資料就可以讓大家瀏覽了！

提供使用者互動介面的動態網站

這種類型的網站可以讓伺服器與使用者互動，常見的例如討論區論壇與留言版，包括一些部落格也都是屬於這類型。這類型的網站是藉由『網頁程式語言』來達成與使用者互動的行為，常見的例如PHP網頁程式語言，配合MySQL資料庫系統進行資料的讀、寫。整個互動可以使用下圖來說明：

圖 20.1-2、動態網站的網頁程式語言與資料庫介面



另一種互動式的動態網頁主要是在用戶端達成的！舉例來說，利用Java scripts將可執行的程式碼(java script)傳送給用戶端，用戶端的瀏覽器如果有提供java script的功能，該程式可在用戶端的電腦上運作。由於程式是在用戶端電腦上執行，因此如果伺服器端程式是惡意的，那用戶端的電腦可能遭到破壞。這也是為啥瀏覽器都將一些危險的java script關閉的原因。

另外一種在用戶端執行的是flash動畫格式，在這種動畫格式內還可以進行程式設計，因此用戶端只要擁有可以執行 flash動畫的軟體，那就可以利用這個軟體來達到互動式的對談。這些都算是動態網站所提供的功能！

從上面的說明你可以知道動態網站是目前比較熱門的，像是個人部落格(blog)就是動態網站之一。而由[圖 20.1-2](http://linux.vbird.org/linux_server/0360apache.php#fig20.1-2)我們也知道要做成這樣的動態網站你必需要有：

* 支援的作業系統：讓所需要的軟體都能夠安裝執行；
* 可運作的WWW伺服器：例如Apache與IIS等WWW伺服器平台軟體；
* 網頁程式語言：包括perl, PHP, JSP, CGI, ASP等等都算是啦！
* 資料儲存之資料庫系統：包括MySQL, MSSQL, PostgreSQL以及甲骨文(Oracle)等等。

LAMP 平台的說明

在整個平台設計上面，目前常見的有兩大系統，一個是Linux上，搭配Apache + MySQL + PHP等而達成，這個系統被稱為LAMP。另一個則是微軟的IIS + MSSQL + ASP (.NET)伺服器：

* Apache: HTTPd經過多次的修訂後，在1995年後發佈Apache(A patch server)！
* MySQL: 傳統的檔案讀取很麻煩，如果只讀取該檔案中的一小部分，系統還是會將整個檔案讀出來，若多人同時讀取同一個檔案時，會造成效能與系統上的問題，所以才有資料庫系統。資料庫是一種特殊格式的檔案，這種檔案必需要透過特殊介面(資料庫軟體)來進行讀寫。由於這個介面針對資料的查詢、寫入做最佳化設計，因此適合多人同時寫入與查詢的工作。針對資料庫的語法有所謂的SQL標準語法，任何根據這種資料檢索語法發展出來的資料庫，就稱為SQL資料庫。MySQL可以透過網頁程式語言來進行讀寫的工作，因此很適合如討論區、論壇等的設計，甚至很多商業網站的重要資料也是透過 MySQL這個資料庫軟體來存取的呢！
* PHP: 按照官方的說法，PHP是一個工具，被用來建立動態網頁，PHP程式碼可以直接在HTML網頁當中嵌入，就像你在編輯HTML網頁一樣的簡單。所以說PHP是一種『程式語言』，這種程式語言可以直接在網頁當中編寫，不需要經過編譯即可進行程式的執行。具有自由軟體、跨平台、容易學習及執行效能高等優點。

20.1.4 https：加密的網頁資料 (SSL) 及第三方公正單位



HTTP協定傳輸資料是以明碼傳送的。雖然多數Internet上的WWW網站所提供的資料可隨意瀏覽，不過一些物流交易網站的資料以及關於你個人的重機密資料當然不能隨意傳送！這時就需要用到https://hostname，透過 SSL加密的機制！

Secure Socket Layer (SSL)

[SSH伺服器](http://linux.vbird.org/linux_server/0310telnetssh.php#ssh)連線機制是利用非對稱的key pair(Public + Private kye)來組成金鑰，透過公鑰加密後傳輸，傳輸到目標主機後再以私鑰來解密，資料以加密的方式傳輸！當瀏覽器端與WWW伺服器端同時支援SSL的傳輸協定時，在連線階段瀏覽器與伺服器會產生那把重要的金鑰！產生金鑰後能夠利用瀏覽器來傳送與接收加密過的重要資料！WWW伺服器要啟動http**s**傳輸協定，而瀏覽器要在網址列輸入https://開頭的網址，兩者才能進行溝通與連線。

Certificate Authorities (CA)

SSL這個機制的問題是：『Public key是伺服器產生且任何人都能取得的』！因public key可讓任何人取得，若被釣魚網站取得並且製作一個很類似你網路銀行的網站，並且騙你輸入帳密！即使你的資料有加密，在釣魚網站伺服器端還是能夠取得你輸入的帳密！這個時候就需要第三方公正單位來幫忙！

CA是一個公認的公正單位，你可以自行產生一把金鑰且製作出必要的憑證資料並向CA單位註冊(註冊就是要錢的意思！)，當用戶端的瀏覽器在瀏覽時，該瀏覽器會主動的向CA單位確認該憑證是否為合法註冊過的，如果是，那該次連線才會建立，如果不是？那麼瀏覽器就會發出警告訊息，告知使用者應避免建立連線。

20.3 Apache 伺服器的進階設定



還有很多可以玩玩的地方，例如個人用戶首頁、虛擬主機以及認證保護的網頁等等。底下我們分別來談一談囉！

20.3.7 虛擬主機的設定 (重要



接下來我們要談的是『主機代管』...是一個稱為虛擬主機～可以讓你的一部Apache看起來像有多個『主站首頁』的感覺！

什麼是虛擬主機 (Virtual Host)

虛擬主機是『讓你的一部伺服器上面，有多個"主網頁"存在，硬體實際上只有一部主機，但由網站網址上來看，似乎有多部主機存在的樣子！』。舉例來說，鳥哥的網站有主要學習網站以及新手討論區，分別在底下的連結：

* 主網站：<http://linux.vbird.org>
* 討論區：[http://phorum.vbird.org](http://phorum.vbird.org" \t "_blank)

兩個連結點會發現不同的資料內容，不過如果用[dig](http://linux.vbird.org/linux_server/0350dns.php#dig)之類的軟體查驗IP，會發現這兩個網址都指向同一個IP！這是虛擬主機的主要功能！讓你的多個主機名稱對應到不同的主網頁目錄(DocumentRoot 參數)，看起來像有多部實際主機的模樣！

20.5 建立連線加密網站 (https)及防砍站腳本



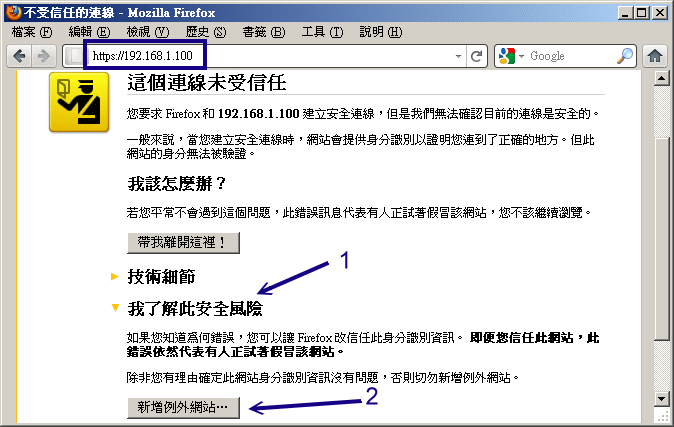
http是明碼傳送資料，而https是加密傳輸！加密的方法是透過SSL，SSL是以openssl軟體來提供的一個加密函式庫。

20.5.1 SSL 所需軟體與憑證檔案及預設的 https



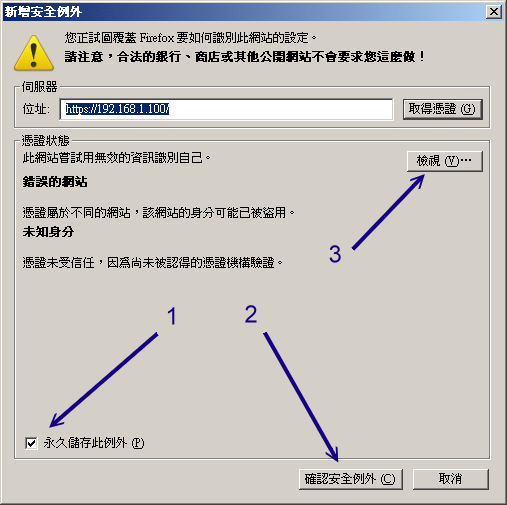
系統預設提供的 https 是長的什麼模樣吧！ 打開你的瀏覽器，輸入https://你的IP來連線看看：

圖 20.5-1、在 firefox 底下看到的 SSL 安全問題圖示



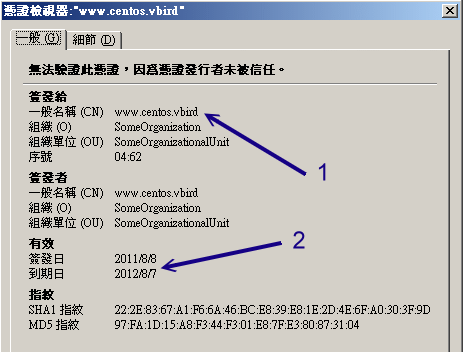
因為這個Apache網站並沒有將此憑證向CA註冊就會出現上述的訊息了！就類似ssh連線時，系統需要你輸入『yes』一樣！要接受憑證後才能夠進行加密的功能。點選上圖中的箭頭 1，此時就會延伸出箭頭 2 的位置，按下去！就會出現如下所示：

圖 20.5-2、在 firefox 底下接受一把私有的憑證所需要的流程



如果你確定這個網站是你自己的可信任網站，那就按下1及2的箭頭處！如果還想要看一下這個網站所提供的相關憑證內容，就按下3箭頭的地方：

圖 20.5-3、在 firefox 底下觀察憑證的詳細內容



由於這個憑證檔案的建置是在第一次啟動Linux時就安裝好了憑證檔，而在CentOS 6.x底下，預設的憑證有效期限為1年，所以就會看到上圖中箭頭2所指的，簽發日到到期日共有一年啊！當你按下關閉後，就能夠看到實際的https://提供的網站內容囉！這就是預設的SSL網站啦！你的重要資訊可以放在這裡～讓資料在網路上傳輸更佳的安全！

20.5.2 擁有自製憑證的 https



建立憑證檔

預設的憑證雖然可讓你順利的使用https，不過憑證的有效日僅有1年而已～所以還是得要自製憑證才行～這個憑證僅是私有WWW網站的用途，並沒有要拿去CA註冊！基本上需要的流程是：

1. 先建立一把private key預備提供給SSL憑證簽章要求所用；
2. 最後建立SSL憑證 (test certificates)。