LAMP 安全設定文件

(Linux、Apache、MySQL、PHP)

TACERT 臺灣學術網路危機處理中心團隊 製

2011/1/6

[此文件主要簡介 LAMP(Linux、Apache、MySQL、PHP)安全設定]

目錄

一、 Introduction

系統與網路防護已成為資訊安全觀念建構之基本概念，在 WWW 平台設計方面，目前常見的有兩大派別：一是 Linux 作業系統搭配 Apache + MySQL + PHP 等所架構而成的環境，這種架構被簡稱為 LAMP。另一派則是微軟的 IIS + MSSQL + ASP (.NET)伺服器。目前不論校內外，就能見度與市佔率而言，還是以 LAMP 最被廣為接受。以下茲就 Linux, Apache, MySQL, PHP 列舉幾項重要且常受攻擊之弱點與其防範方法，供系統管理者參考之用。

二、 Linux 安全設定

（一） 建立完善的登入密碼規則限制

1. 說明：因為密碼常是駭客嘗試入侵的前哨站，而密碼安全也是保護系統最基本的層面，所以建立良好主機的密碼規則是非常重要的。
2. 相關設定：

(1)	可利用"Crack" and "John the Ripper"確認密碼的強度
(2)	採用 Shadow passwords
(3)	以 chattr 將/etc/passwd 及/etc/shadow 做成不可變更的檔案

3. 詳細說明：請查閱參考資料[8]的 1.建立完善的登入密碼規則限制章節、參考資料[9]的 6.8. Shadow Passwords.及 6.9. "Crack" and "John the Ripper"章節。

（二） 完善的主機權限設定

1. 說明：如果意圖破壞的人是我們系統上具有可以登入的使用者，不安全的權限設定可能會讓一般身份的使用者可以很輕易的取得系統管理員的執行權限。
2. 詳細說明：請查閱參考資料[8]的 2.完善的主機權限設定章節、參考資料[11]的 User Account Security 章節。

（三） 升級與修補套件漏洞、及移除危險套件

1. 說明：確保服務套件的安全性，避免安全風險，以達到基本保護的效果。
2. 相關設定：

(1)	不啟動不安全或不需要的服務
(2)	若是需要啟動可能不安全的服務，應該限制其使用網域

3. 詳細說明：請查閱參考資料[8]的 3.升級與修補套件漏洞、及移除危險套件章節。

（四） 取消伺服器軟體較危險的功能

1. 說明：每個伺服器軟體都有自己開發的功能，有些功能很強大，強大到可以讓 Client 端取得 root 的權限來操作，但也因此產生安全上的疑慮，所以要避免不合法的使用者以 root 帳號對密碼進行暴力破解，登入系統。

2. 相關設定：

(1) 建議取消 SSH, FTP, Telnet 等軟體的 root 登入權限
(2) 改用 Openssh 取代明碼的連線方式

3. 詳細說明：請查閱參考資料[8]的 4.每項系統服務的安全設定項目章節、參考資料[9]的 6.4. ssh (Secure Shell) and stelnet 章節、參考資料[11]的 Working with OpenSSH 章節、參考資料[12]。

（五） TCP_Wrappers 的基礎防火牆設定

1. 說明：TCP_Wrappers 主要以『服務, services』為主的抵擋方針，我們可以利用 TCP_Wrappers 訂定某些比較危險的服務僅針對內部網路開放，以減少安全風險。

2. 相關設定：

(1) 確認是否已經安裝 TCP_Wrappers
rpm -q tcp_wrappers
(2) 安裝 TCP_Wrappers

3. 詳細說明：請查閱參考資料[8]的 5.TCP_Wrappers 的基礎防火設定章節、參考資料[9]的 8.2. System services and tcp_wrappers、參考資料[13]的服務的防火牆管理 xinetd, TCP Wrappers。

（六） iptables 的防火規則設定

1. 說明：iptables 可以經由 TCP 的封包 Header 資料來進行分析的工作，以設定防火牆的安全規則，如此則可以抵擋掉大部分的不受歡迎的 TCP 封包。

2. 詳細說明：請查閱參考資料[8]的 6.iptables 的防火規則設定章節、參考資料[9]的 8.13. Netfilter - Linux Kernel 2.4.x Firewalling 章節。

（七） 更多

關於 Linux 的安全還包括了很多其他的部分，若更深入瞭解的話，可以查閱以上參考資料的其他章節。

三、 Apache 安全設定

（一） 確保 Apache 是以特定的 user 帳號執行 service

1. 說明：確認 Apache 是以低權限的 user 帳號執行，避免使用 root 權限。

2. 目的：避免被駭客破解後利用最高權限進行攻擊。

3. 相關設定：

```
User apache
Group apache
```

4. 詳細說明：請查閱參考資料[4]的 Make sure apache is running under its own user account and group 章節。

（二） 嚴格控管 ServerRoot 目錄的權限

1. 說明：一般而言，Apache 是由 the root user 啟動，然後才將權限換成管理的 user。這樣一來，就都是 root 執行的指令，所以我們要確認這些指令及設定不會被 non-root user 所修改。
2. 目的：確保 non-root user 無法修改 Apache 的目錄及檔案以執行其他系統檔案或改寫 log 檔。
3. 相關設定：

For example, if you choose to place ServerRoot in /usr/local/apache then it is suggested that you create that directory as root, with commands like these:

```
mkdir /usr/local/apache
cd /usr/local/apache
mkdir bin conf logs
chown 0 . bin conf logs
chgrp 0 . bin conf logs
chmod 755 . bin conf logs
```

It is assumed that /, /usr, and /usr/local are only modifiable by root. When you install the httpd executable, you should ensure that it is similarly protected:

```
cp httpd /usr/local/apache/bin
chown 0 /usr/local/apache/bin/httpd
chgrp 0 /usr/local/apache/bin/httpd
chmod 511 /usr/local/apache/bin/httpd
```

You can create an htdocs subdirectory which is modifiable by other users -- since root never executes any files out of there, and shouldn't be creating files in there.

4. 詳細說明：請查閱參考資料[1]的 Permissions on ServerRoot Directories 章節。

（三） 控管可存取 Apache Server 的網段或 IP

1. 說明：如果我們知道將會存取 Apache Server 的網段或 IP，我們可以在 Apache 設定中，對這些會存取 Apache 的 clients 加以控管。
2. 目的：可避免非法的 IP 存取 Apache Server。
3. 相關設定：

(3) 修改/usr/local/etc/apache/httpd.conf

- By 網段

```
Order Deny,Allow
Deny from all
Allow from 176.16.0.0/16
```

- By IP

```
Order Deny,Allow
```

```
Deny from all
Allow from 127.0.0.1
```

4. 詳細說明：請查閱參考資料[4]的 Restricting Access by IP 章節。

（四） 禁止 Apache 存取 Web 目錄外的其他文件

1. 說明：我們並不想讓 Apache 可以去存取除了 web root 目錄外的其他檔案，所以假設網站的所有內容都放在同一目錄下。
2. 目的：避免 Apache 存取其他系統或重要檔案。
3. 相關設定：

(4) 修改/usr/local/etc/apache/httpd.conf

```
<Directory />
   Order Deny,Allow
   Deny from all
   Options None
   AllowOverride None
</Directory>
<Directory /web>
   Order Allow,Deny
   Allow from all
</Directory>
```

4. 詳細說明：請查閱參考資料[4]的 Ensure that files outside the web root are not served 章節。

（五） 關閉 Apache 顯示目錄結構列表

1. 說明：Apache 預設會顯示目錄列表，包含目錄及檔案的資訊。我們可以將該功能關閉。
2. 目的：避免完整的目錄結構資訊被惡意人士取得及利用。
3. 相關設定：

(1) 修改/usr/local/etc/apache/httpd.conf

(2) 修改 Directory 標籤的 Options 屬性

```
Options -Indexes
```

4. 詳細說明：請查閱參考資料[4]的 Turn off directory browsing 章節。

（六） 關注 Log 記錄

1. 說明：雖然 Log 檔案是記錄已經發生過的事件，但仍可讓我們知道該 Server 遭受到了哪些攻擊，可協助我們判斷是否需要提高目前的安全等級。
2. 目的：確認 Server 遭受到了哪些攻擊，以供判斷目前的安全等級是否需要提高。
3. 詳細說明：請查閱參考資料[1]的 Watching Your Logs 章節。

（七） 更新 Apache 至最新版本

1. 說明：雖然 The Apache HTTP Server 在安全方面有良好的記錄，而且還有一個關注相關安全議題的開發者社群，但仍會有 released 後才發現的問題。
2. 目的：升級 Apache 以修補漏洞。
3. 相關設定：

(1) 訂閱 Apache HTTP Server Announcements List，以查看是否有更新的版本釋出或安全性更新。
(2) 升級至最新版本及安裝安全性更新。

4. 詳細說明：請查閱參考資料[1]的 Keep up to Date 章節。

（八） 關閉 Apache Server 所提供 SSI 機制的 exec 功能

1. 說明：Server Side Includes(SSI)代表了幾個潛在安全風險的 Server administrator。第一個風險是增加 Server 的負擔，因為所有的 SSI-enabled files 不論檔案內是否有 include 任何 SSI 的指令，都要經過 Apache 的 parse，如果是在 a shared server environment 上，就會變得很明顯。SSI 檔案也會發生和 CGI scripts 相同的風險，SSI 可以透過 exec 指令來以 Apache 的 user 和 group 權限執行任何 CGI script 或程式。
2. 目的：避免透過 exec 指令來以 Apache 的 user 和 group 權限執行任何 CGI script 或程式。
3. 相關設定：

(1) 啟用 suexec 功能。
(2) 關閉 SSI 功能。

4. 詳細說明：請查閱參考資料[1] Server Side Includes 章節。

（九） 隱藏 Apache 的版本和其他敏感資訊

1. 說明：Apache 會顯示我們所執行的 Apache 版本、作業系統類型及版本，甚至是安裝在 Server 上的 Apache Modules，而駭客可以利用這些資訊及弱點進行攻擊。
2. 目的：避免敏感資訊被駭客取得及利用。
3. 相關設定：

(1) 修改 httpd.conf

```
ServerSignature Off
ServerTokens Prod
```

4. 詳細說明：請查閱參考資料[4] Hide the Apache Version number, and other sensitive information.章節。

（十） 更多

關於 Apache 的安全還包括了很多其他的部分，若更深入瞭解的話，可以查閱以上參考資料的其他章節。

四、 MySQL 安全設定

（一） root 帳號需設定強密碼

1. 說明：因為 root 為擁有所有權限以及可以做任何事的 superuser accounts，而其初始密碼為 empty，所以在預設的情況下，任何人都能在沒有密碼的情況下連接 MySQL server。
2. 目的：避免其他人在沒有密碼的情況下連接 My SQL server，並擁有最大權限。
3. 相關設定：

```
shell> mysql -u root
mysql> SET PASSWORD FOR 'root'@'localhost' = PASSWORD('newpwd');
mysql> SET PASSWORD FOR 'root'@'127.0.0.1' = PASSWORD('newpwd');
mysql> SET PASSWORD FOR 'root'@'::1' = PASSWORD('newpwd');
mysql> SET PASSWORD FOR 'root'@'host_name' = PASSWORD('newpwd');
```

4. 詳細說明：請查閱參考資料[2] 2.2. Securing the Initial MySQL Accounts 章節。

（二） Anonymous 帳號的刪除/設定密碼

1. 說明：因為 Anonymous users 使用的是 empty 的使用者帳號，而且沒有密碼，所以任何人都能使用 Anonymous users 去連接 MySQL server。
2. 目的：避免任何人都能使用 Anonymous users 去連接 MySQL server。
3. 相關設定：
   - 修改密碼

```
shell> mysql -u root -p
Enter password: (enter root password here)
mysql> UPDATE mysql.user SET Password = PASSWORD('newpwd')
    ->       WHERE User = '';
mysql>
```

   - 刪除 Anonymous 帳號

```
shell> mysql -u root -p
Enter password: (enter root password here)
mysql> DROP USER ''@'localhost';
mysql> DROP USER ''@'host_name';
```

4. 詳細說明：請查閱參考資料[2] 2.2. Securing the Initial MySQL Accounts 章節。

（三） 刪除其他不使用的資料庫及使用者帳號

1. 說明：My SQL 預設的情況下，以 test_開頭的資料庫允許任何使用者存取。
2. 目的：避免 MySQL 預設而產生的其他帳號和資料庫不僅沒有使用

到，反而還對資料庫產生威脅。

3. 相關設定：

```
mysql> drop database test;
mysql> use mysql;
mysql> delete from db;
mysql> delete from user where not (host="localhost" and user="root");
mysql> flush privileges;
```

4. 詳細說明：請查閱參考資料[3] 4.4 Remove default users/db 章節。

（四） 改變預設管理者的帳號

1. 說明：改變管理者的預設帳號(root)也是建議的，因為會增加管理者密碼被暴力破解的難度，侵入者不僅要猜測密碼，還要再猜測使用者名稱。
2. 目的：可以避免針對管理者密碼的暴力攻擊。
3. 相關設定：

```
mysql> update user set user="mydbadmin" where user="root";
mysql> flush privileges;
```

4. 詳細說明：請查閱參考資料[3] 4.5 Change admin name 章節。

（五） 停用 LOAD DATA LOCAL INFILE 指令

1. 說明：LOAD DATA 可以載入位於 the server host 上的檔案，或是利用 LOCAL 關鍵字來載入位在 the client host 的檔案。
2. 目的：避免兩種風險：(1)將導致 server 可以存又任何 the client user 已經 read access 過的，位於 the client host 上的檔案(2)使用者可以使用 LOAD DATA LOAL 讀取任何 the Web server process 已經 read access 的檔案(假設使用者可以對 the SQL server 下任何指令)。
3. 相關設定：

(1) 將以下參數加入至 my.cnf 中的[mysqld]section 中

```
set-variable=local-infile=0
```

4. 詳細說明：請查閱參考資料[2] 1.5 Security Issues with LOAD DATA LOCAL 章節、參考資料[5] 3. Disable the use of LOCAL INFILE 章節。

（六） 禁止遠端連接 MySQL

1. 說明：MySQL 預設會監聽 3306/tcp port，根據假設的前提下，資料庫應該只會被本地端的 PHP 存取，所以我們可以禁止遠端連接 MySQL。
2. 目的：將不需要的功能關閉，避免潛在威脅。
3. 相關設定：

(1) 修改/chroot/mysql/etc/my.cnf

增加 skip-networking 參數至[mysqld] section

(2) 若基於某些原因，需對資料庫進行 remote access，則必須使用 SSH，詳情請見參考資料[2] 4.9. Connecting to MySQL Remotely from Windows with SSH 章節

4. 詳細說明：請查閱參考資料[3] 4.1 Disable remote access 章節。

（七） 控制資料庫存取權限

1. 說明：資料庫管理者可以使用二個帳號(root 和另一個)，root 帳號不給予存取資料的權利，純粹進行資料庫維護的工作，而一般的 user 帳戶可以存取資料，例如 web server 的 user id 可以給予執行 “select\update\insert\delete” queries 以及執行 stored procedures 的權限。只有 administrator 帳號可以授與 SUPER / PROCESS /FILE 的權限以及對資料庫的存取權限。
2. 目的：權責明確的區分，只給予資料庫的使用者適當的權限。
3. 相關設定：

依下列步驟檢查資料庫 users 的存取權限是否適當：

```
mysql> use mysql;
```

[Identify users]

```
mysql> select * from users;
```

[List grants of all users]：對每位 user 進行權限的確認

```
mysql> show grants for ‘root’@’localhost’;
```

4. 詳細說明：請查閱參考資料[5] 8. Lower database privileges 章節。

（八） 限制”SHOW DATABASES”的權限

1. 說明：預設情況下，每個人都可以以此來作為攻擊資料庫前的資訊蒐集，如：竊取資料等。
2. 目的：
3. 相關設定：

(1) 限制”SHOW DATABASES”的權限

- 修改/etc/my.cnf

加上“ --skip-show-database”

- 只授予適當的使用者”SHOW DATABASES”的權限

(2) 若想關閉”SHOW DATABASES”的指令，將以下指令加入到 /etc/my.cnf 的[mysqld]中。

```
[mysqld]
skip-show-database
```

4. 詳細說明：請查閱參考資料[5] 8. Lower database privileges 章節、[16] --skip-show-database 章節。

（九） 修改資料庫檔案目錄的存取權限

1. 說明：假設 MySQL 安裝在/chroot/mysql 目錄下，所以相對的資料庫文件就是在/chroot/mysql/var 目錄下，所以我們要限制對這些目錄的存取權限。
2. 目的：避免未經授權的 user 將資料庫 copy 帶走。
3. 相關設定：

The access rights to the above directories should be set as follows:

```
chown -R root:sys /chroot/mysql
chmod -R 755 /chroot/mysql
chmod 1777 /chroot/mysql/tmp
```

4. 詳細說明：請查閱參考資料[3] 3.3 Set access rights 章節。

（十） 歷史記錄的處理

1. 說明：因為所有 MySQL 執行過的的指令會被記錄在 the MySQL history file (*~/.mysql_history*)，裡面可能包含了密碼等敏感性資訊。
2. 目的：避免駭客取得敏感性資訊。
3. 相關設定：

(1) 改變存放歷史記錄的檔案(建議)
設定環境變數 MYSQL_HISTFILE
(2) 刪除歷史記錄

```
Set the MYSQL_HISTFILE variable to /dev/null.
cat /dev/null > ~/.mysql_history
```

4. 詳細說明：請查閱參考資料[3] 4.6 Remove history 章節、參考資料[5] 11. Remove History 章節、參考資料[16]2.14. Environment Variables 及 4.5.1.3. mysql History File 章節。

（十一） 可考慮使用 chroot 來控制 mysql 的目錄

1. 說明：chroot 是一個用來改變一個正在執行的 process 及其 children 所參考的 root 目錄，可以限制存取檔案、程式的範圍。
2. 目的：避免使用者取得其他敏感性資訊(例：/etc/passwd)。
3. 相關設定：

(1) 確保 mysql 存放在一個 chroot 環境專用的目錄，例：

```
/chroot/mysql
```

(2) 為了方便資料庫管理工具的使用方便，必須修改 MySQL configuration file 中的[client]區塊

```
[client]
socket = /chroot/mysql/tmp/mysql.sock
```

註：在 chroot 下的 MySQL 和 PHP 間的溝通也必須要設定 hard link。(查閱參考資料[3])

4. 詳細說明：請查閱參考資料[3] 5. Communication between PHP and MySQL 章節、參考資料[5] 10. Change the root directory 章節。

（十二） 更多

關於 MySQL 的安全還包括了很多其他的部分，若更深入瞭解的話，可以查閱以上參考資料的其他章節。

五、 PHP 安全設定

（一） include 限制

1. 說明：include 功能可分為 Local include 及 Remote file include 兩種情形。Local include 讓駭客能夠將存有敏感資訊的檔案 include 進 PHP scripts，以檢視這些資訊(例如：/etc/passwd 等)；而 Remote file include 可以讓駭客不需登入我們的系統，就可直接使用 remote inclusion method 上傳到我們的 server 並執行 scirpt，使我們的系統產生弱點。
2. 目的：避免 Local include attacks 及 Remote file include attacks。
3. 相關設定：

(1) 控制 PHP 能存取的本地目錄
修改 php.ini

```
include_dir
```

(2) 關閉開啟遠端檔案、允許 include 遠端檔案的功能
修改 php.ini

```
allow_url_fopen = off
allow_url_include = off
```

4. 詳細說明：請查閱參考資料[6] Restricting Includes 章節、參考資料[15] 4.9 Don’t use PHP allow_url_fopen 章節。

（二） 隱藏 PHP 版本

1. 說明：一般而言，隱藏是最脆弱的確保安全型式。
2. 目的：隱藏 PHP 資訊以減緩駭客查覺系統弱點的速度。
3. 相關設定：

(1) 修改 php.ini

```
expose_php = off
```

(2) 模糊 PHP code 成未知的格式
修改 Apache 的 httpd.conf

```
# Make PHP code look like unknown types
AddType application/x-httpd-php .bop .foo .133t
```

(3) 模糊 PHP code 成 HTML 格式
修改 Apache 的 httpd.conf

```
Make all PHP code look like HTML
AddType application/x-httpd-php .htm .html
```

4. 詳細說明：請查閱參考資料[14] Hiding PHP 章節。

（三） 關閉 register_globals

1. 說明：register_globals 指令在 PHP 版本 4.2.0 版本後預設為關閉，雖然它不是安全漏洞，但有安全的風險，所以建議關閉 register_globals。
2. 目的：避免駭客可以改寫設定變數以取得系統的存取權限。
3. 相關設定：

(1) 修改 php.ini

```
register_globals = off
```

4. 詳細說明：請查閱參考資料[6] Disable Register Globals 章節、[7] 1.3 Register Globals 章節、[14] Using Register Globals 章節。

（四） 關閉 magic_quotes_gpc

1. 說明：因為 PHP 不再支援 magic_quotes，而 magic_quotes 原本存在的原因是基於防止 SQL Injection 的考量。
2. 目的：因為 magic_quotes 並非所有 escaped data 都會寫入資料庫，這樣一來會導致 performance loss，基於效能的原因，所以建議關閉 magic_quotes_gpc。
3. 相關設定：

(1) 修改 php.ini

```
; Magic quotes for incoming GET/POST/Cookie data.
magic_quotes_gpc = Off

; Magic quotes for runtime-generated data, e.g. data from SQL, from exec(), etc.
magic_quotes_runtime = Off

; Use Sybase-style magic quotes (escape ‘ with ‘’ instead of \’).
Magic_quotes_sybase = Off
```

(2) 若無法對 the server configuration 存取，也可修改.htaccess 檔案

```
php_flag magic_quotes_gpc Off
```

4. 詳細說明：請查閱參考資料[14] Magic Quotes 章節。

（五） 避免資訊的揭露

1. 說明：出現在 the web application’s user interface 的錯誤訊息中，會含有 PHP 當前路徑或使用者的 SQL 語法等跟系統有關的資訊。
2. 目的：避免駭客得知我們系統的目錄結構、資料庫名稱等更多資訊。
3. 相關設定：

(1) 關閉錯誤訊息在 user interface 的顯示
修改 php.ini

```
display_errors =Off
```

註：雖然錯誤訊息顯示的功能關閉了，但還是能從 PHP 的 log 記錄中得知；或是在開

發 PHP 時，才將功能打開，開發完成後將功能關閉。

4. 詳細說明：請查閱參考資料[6]Preventing Information Disclosure 章節、[7]1.5 Error Reporting 章節、[14] Error Reporting 章節。

（六） 限制檔案上傳

1. 說明：駭客會利用 PHP 的設定，試圖注入他們的 PHP scripts，所以如果 PHP 的開發者需要使用檔案上傳的功能，應該要設定存放目錄為獨立的另一資料夾，此時便要修改 upload_tmp_dir 裡的參數。
2. 目的：限制上傳的檔案大小通常不是為了安全的目的，而是為了管理 Server 的 PHP loading。
3. 相關設定：

(1) 開啟檔案上傳功能、設定存放目錄的路徑

```
file_uploads = On
load_tmp_dir=存放目錄的路徑
```

(2) 限制上傳的檔案大小，例：限制上傳檔案大小為 2M
修改 php.ini

```
upload_max_filesize = 2M
```

4. 詳細說明：請查閱參考資料[6]Restrict File Uploads 章節。

（七） Sessions 保護

1. 說明：駭客可能會利用注入 JavaScript 到網頁的方式來竊取 cookies，因此可以藉由設定 session.cookie_httponly 參數為 On(指定其值為 1)。若我們的 PHP 開發者需要利用 JavaScript 來存取 session cookies，請勿開啟這個選項。
2. 目的：避免駭客利用 JavaScript 存取 PHP 的 session cookies。
3. 相關設定：

(1) 限制 JavaScript 存取 cookies
修改 php.ini

```
session.cookie_httponly=1
```

4. 詳細說明：請查閱參考資料[6] Protect Sessions 章節、[7] 4. Sessions 章節。

（八） 更新 PHP 至最新版本

1. 說明：PHP 的每個新版本通常都會包含加強安全性以及漏洞的修復等主要和次要的調整，而這些都會影響到我們系統的整體安全性和穩定性。
2. 目的：升級 PHP 以修補漏洞。
3. 詳細說明：請查閱參考資料[14]的 Keeping Current 章節。

（九） 更多

關於 PHP 的安全還包括了很多其他的部分，若更深入瞭解的話，可以

查閱以上參考資料的其他章節。

六、 結論

資安攻防訊息更新日新月異，世上也沒有百分百安全的系統，身為系統管理員務求迅速更新，跟進腳步，為免成為網路上不法份子的受害對象，本文件因筆者能力有限，僅能列舉一部份常見問題，內容尚有不足，其餘問題需仰賴系統管理員不斷精進、自我充實，方能提供使用者一個安全層級較高的開發環境。

七、 參考資料

[1]Apache HTTP Server Version 2.2 – Security Tips
http://httpd.apache.org/docs/current/misc/security_tips.html
[2]Security in MySQL
http://dev.mysql.com/doc/mysql-security-excerpt/5.5/en/index.html
[3]Securing MySQL: step-by-step | Symantec Connect
http://www.symantec.com/connect/articles/securing-mysql-step-step
[4]20 ways to Secure your Apache Configuration
http://www.petefreitag.com/item/505.cfm
[5]MySQL Security Best Practices (Hardening MySQL Tips)
http://www.greensql.net/publications/mysql-security-best-practices
[6]PHP Security Concepts
http://docs.cpanel.net/twiki/bin/view/AllDocumentation/WHMDocs/PhpSecurityConcepts
[7] PHP Security Guide
http://phpsec.org/projects/guide
[8]鳥哥的 Linux 私房菜 – 主機防護計畫
http://freedom.mingann.info/vbird/linux_server/0240network-secure-1.htm#secure_basic
[9]Linux Security HOWTO
http://tldp.org/HOWTO/Security-HOWTO/
[10]Implement password security in Linux
http://articles.techrepublic.com.com/5100-10878_11-1049408.html
[11]Linux Online's Course for Advanced Users
http://www.linux.org/lessons/advanced/
[12]鳥哥的 Linux 私房菜 – 以 Openssh 取代 Telnet 的連線方式
http://linux.vbird.org/linux_security/old/08telnet-ssh.php
[13]鳥哥的 Linux 私房菜 –服務的防火牆管理 xinetd, TCP Wrappers
http://linux.vbird.org/linux_basic/0560daemons.php
[14]PHP: Security – Manual

http://php.net/manual/en/security.php
[15]Security Checklist 2 – Hosting and Server Setup
http://docs.joomla.org/Security_Checklist_2_-_Hosting_and_Server_Setup
[16]MySQL 5.5 Reference Manual
http://dev.mysql.com/doc/refman/5.5/en/environment-variables.html