**網站安全性**

如果以為對主機裝設掃毒軟體或防火牆，那就對安全性的真正意義完全不了解。

**Register Globals**

Php.ini組態設定檔設定register\_globals為on時，程式檔可接受所有種類的變數，包含ENVIRONMENT，GET，POST，COOKIE，與SERVER變數。

因為php不需要事先初始變數數值，所以register\_globals為on時，可以從HTTP請求中設定變數。



或是從http請求中區更改session值(範例檔: sessionMain.php)



register\_globals為off時，不表示絕對安全，對於任何傳送的資料還是要經過仔細的驗證，變數也要事先設定初始值。目前php6已移除register\_globals。

**安全模式**

Php.ini的safe\_mode選項設定為on，用來限制檔案的存取，限制環境變數的存取，與控制外部程序的執行。

伺服器以單一系統使用者模式進行，因此系統使用者必須能夠存取每個使用者的檔案，這表示網站伺服器上執行的任何程式檔案，都能夠存取每個使用者的檔案

PHP在安全模式中執行時，只能限制PHP檔案，因此外部應用程式應該要放在安全的資料夾(使用safe\_mode\_exec\_dir=路徑，來指定資料夾的路徑)，不要讓外面使用者執行。使用PHP程式黨執行檔案操作時，檔案擁有者user id需與PHP程式檔擁有者user id相同。safe\_mode在Php6已移除不再使用。

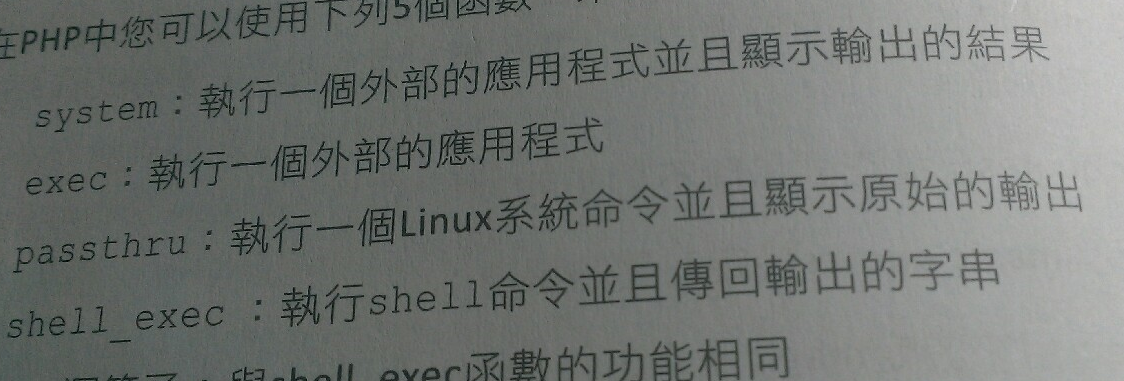
**Magic Quotes**

Php.ini設定magic quotes為on，可將輸入的資料自動加入一個\作跳脫動作，PHP6已移除不再使用。

**攻擊種類**

1. **命令植入攻擊**

Php中可使用system、exec、passthru、shell\_exec、‵‵運算子，來執行外部的應用程式或命令。



攻擊實例

範例檔位於commandIniection資料夾

防禦方法:

1. 盡量不要執行外部應用程式或指令
2. 使用自訂的函式庫取代外部應用程式或命令功能
3. 使用escapeshellarg函數處理命令參數
4. 使用safe\_mode\_exec\_dir指定可執行檔案路徑
5. 事先列出可以提供給system，eval等函數的參數

escapeshellarg函數會將引取參數或命令結數字元跳脫如:單引號用\’取代，雙引號用\”、分號用\;取代。

1. **eval植入攻擊**

eval函數會將輸入的字串當PHP程式碼執行。

攻擊實例

範例檔位於evalIniection資料夾

防禦方法

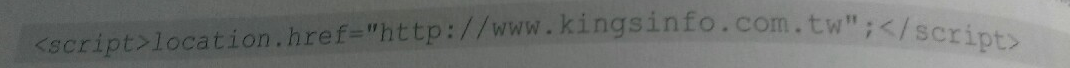
如上、與命令植入攻擊相同。

1. **用戶端植入攻擊**

將可執行的指令碼植入表單、圖片、超連結物件中，當打開這些物件，就會被這些指令碼攻擊。

**攻擊實例**

例如留言板功能是供人留言觀看的地方，留言資料都會使用資料庫存取，如果在留言板輸入底下javascript碼並儲存(範例檔位於scriptInjection資料夾)



資料庫存取後，等受害人瀏覽留言板時，網頁就會讀取資料庫資料並執行這串指令碼，將受害人網址轉移到充滿病毒的網站。

防禦方法

1. 使用strip\_tags函數去除字串中的HTML與php標記
2. 使用htmlspecialchars函將特殊字元轉換成HTML字元碼

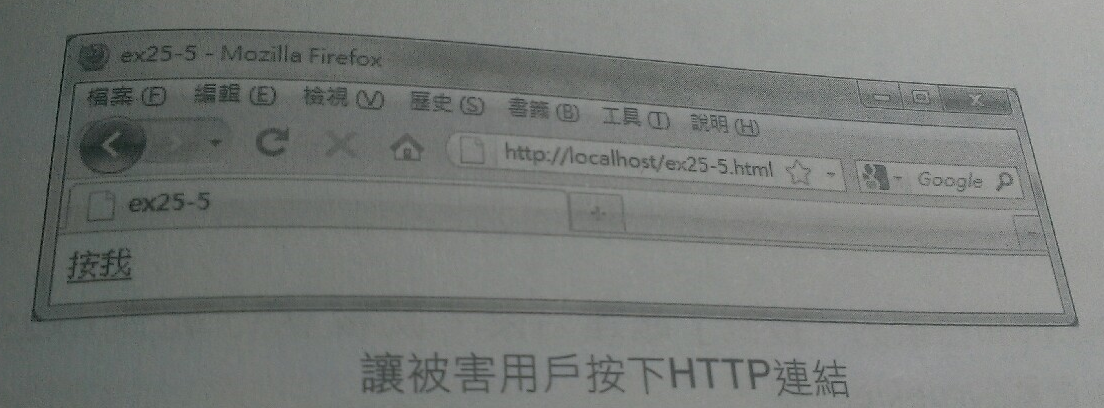
**4.跨網站指令碼攻擊**

XSS是Cross Site Script的縮寫，目前WEB應用程式都是利用cookie儲存用戶的帳號密碼等資料，當網站的應用程式對用戶入的資料沒做檢驗，駭客就能將攻擊的JS程式碼加在http連結後。WEB應用程式直接執行httP連接，讓攻擊的JS程式碼啟動。

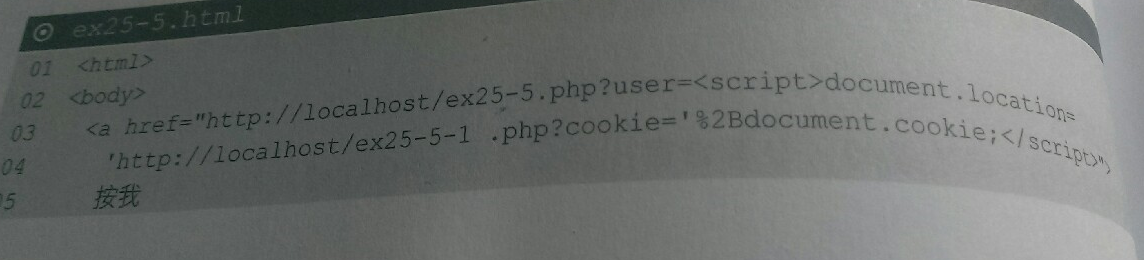
攻擊實例

範例檔位於XSS資料夾

駭客利用電子郵件或其他方式誘使被害人點擊連結



當點擊後會執行此連結背後所藏的JS程式碼



此程式碼會將用戶的cookie資料傳回駭客的網站，駭客可從他的伺服器記錄檔(appserv為例，記錄檔在appserv\apache2.2\logs資料夾中的access.log)，取得任何http請求的資料。

防禦方法

1. 使用htmlspecialchars函數轉換特殊字元為html字元碼。

**5.SQL植入攻擊**

在瀏覽器的表單資料，WEB應用程式會利用$\_POST變數來存，然後在SQL陳述式與POST變數做結合，駭客就能在輸入的表單資料加入特殊程式碼，來改變原來的SQL陳述式。

攻擊實例:

範例檔位於SQLInjection資料夾

下圖為類似檢查帳號密碼的入口網頁



駭客只要帳密都輸入

Sql陳述式就會變成:



因為結果為True，所以這個陳述式不會檢查帳密輸入的值，而是直接查詢member資料表所有的紀錄。

防禦方法

檢查陳述式中的變數資料是否是正確格式

1. 如果資料確定是整數，使用intval函數將資料轉為整數，因為id通常是由mysql資料庫產生的整數型態數字，所以使用intval函數將id值轉為整數，避免id值被串改。
2. 如果確定是浮點數，就使用doubleval,將資料轉換成浮點數。
3. 如果確定資料是字串，就是用addslashes函數將單引號，雙引號，\，以及null字元加上反斜線。

**6.跨網站請求偽造攻擊**

Cross Site Request Forgeries(CSRF)，駭客偽造被害用戶的http請求，然後讓被害用戶在不注意情況下將這個請求傳送給有CSRF漏洞的網站。

因為http連結是擁有特定合法權限的被害用戶自己發出，因此漏洞的網站將此請求視為正常的http請求而執行他。

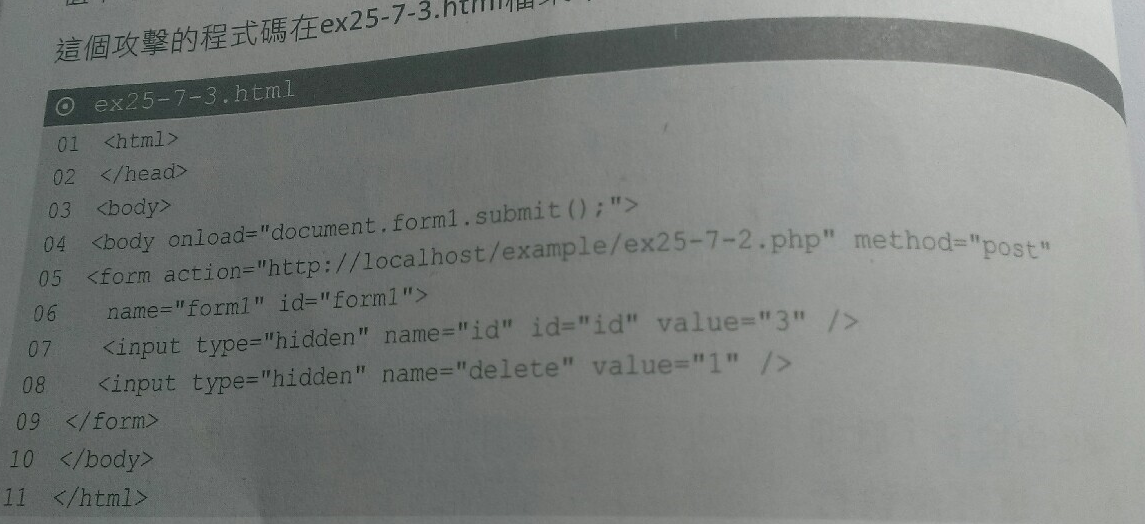
攻擊實例

範例檔位於CSRF資料夾

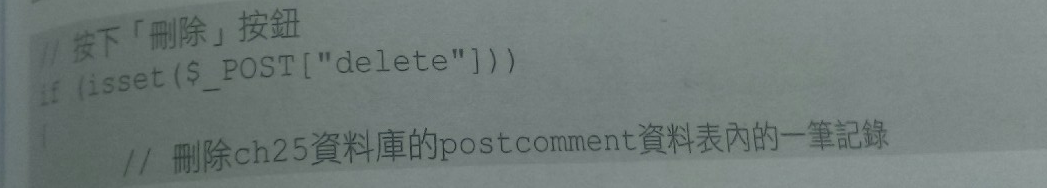
下圖為一留言板，並且有能刪除自己的留言功能

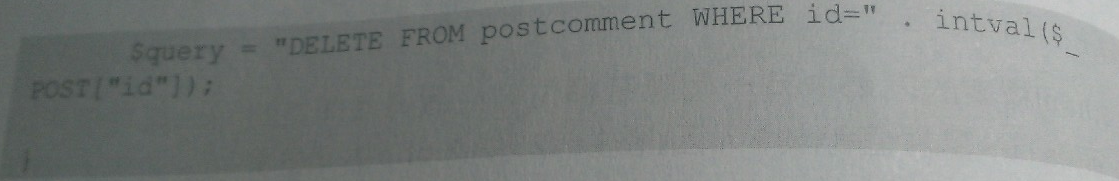


當駭客利用電子郵件或惡意方法讓被害用戶誤擊連結導入攻擊網頁如下圖:



被害用戶點擊連結後，進入上圖的攻擊網頁，在一開始就會載入submit動作將delete，與id參數傳回留言板的PHP程式檔。

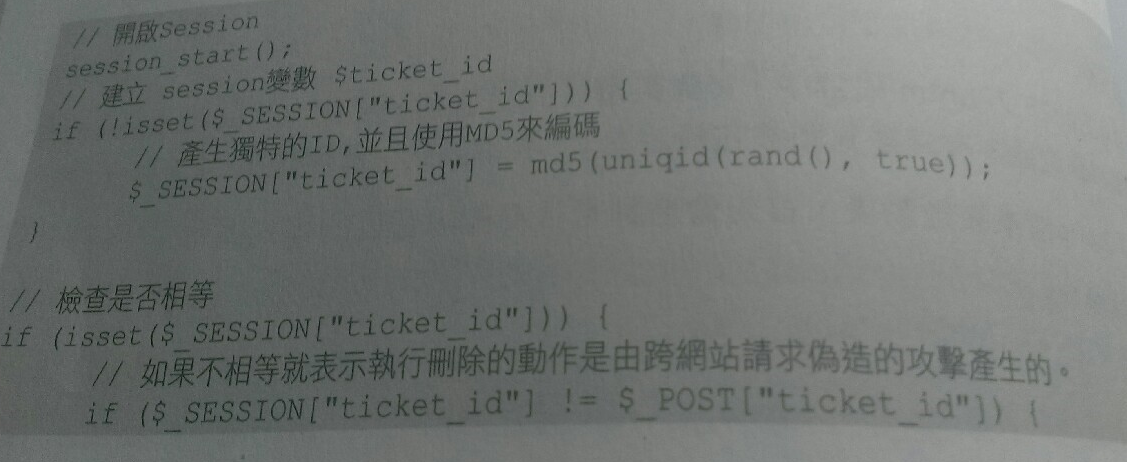


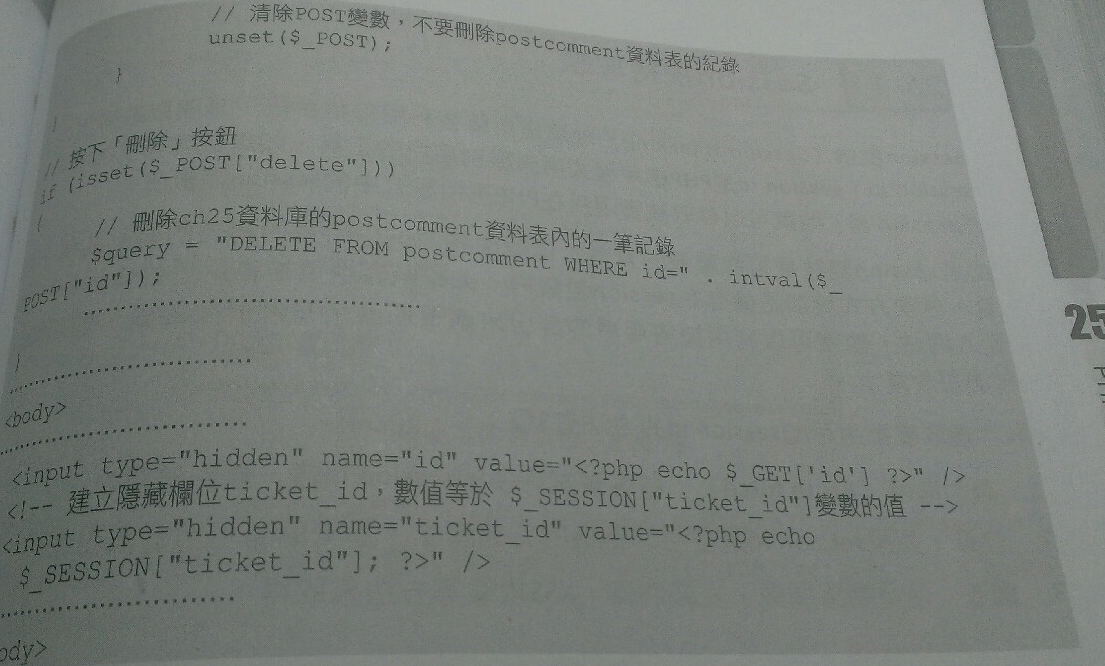


從攻擊頁面傳來的delete與id參數直接執行了刪除的動作，而被害用戶還毫不知情發生什麼事。

防禦方法

在表單內建立一隱藏變數以及一個session變數，檢查兩個變數是否相等，來判斷是否由同一個網頁呼叫。





當PHP檔案第一次被呼叫時，就會產生一個session變數$ticket\_id。並在隱藏欄位中存session值，所以只要判斷$\_session[ticket\_id]與$\_post[ticket\_id]是否相等，如果不相等就表示刪除的動作是由跨網站請求伪造所產生的。

**7.Session挾持攻擊**

Session變數是重要的資料，很多網站都將資料儲存在session變數內，session id是PHP應用程式用來識別用戶的工具，駭客如果取得了用戶session id，就可以利用被害用戶身分做許多對用戶有害的動作。

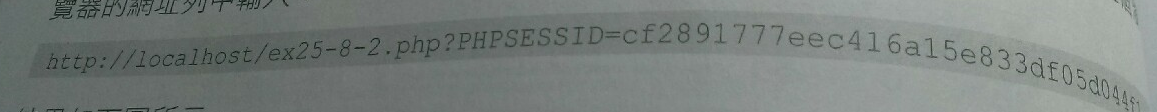
駭客取的session id方法:

1. 暴力破解:駭客嘗試許多session id值，直到破解
2. 計算:如果session是使用非隨機方式產生，有可能被計算出來
3. 竊取:使用網路攔截、病毒、XSS方式取得

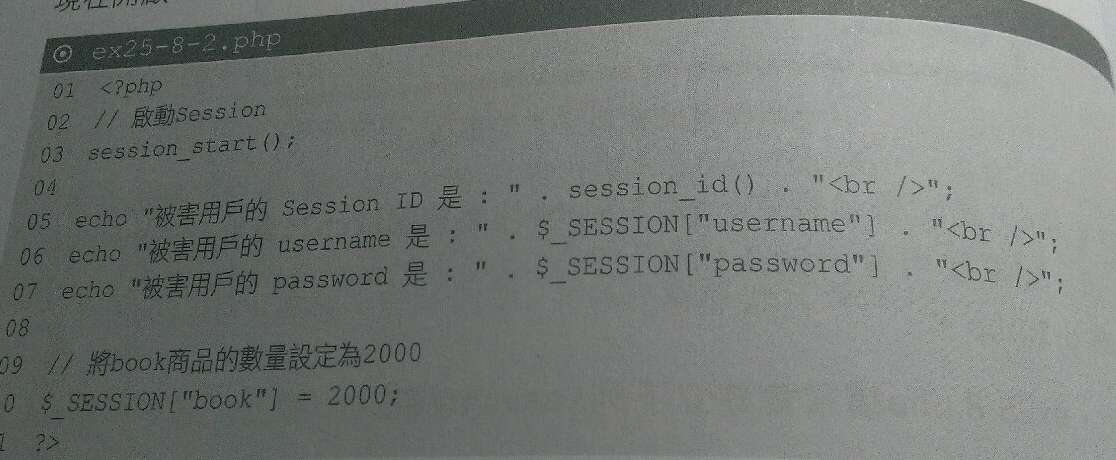
攻擊實例

範例檔位於sessionInjection資料夾

假設駭客取得了登入者的session id，他可利用下面URL進入駭客寫的PHP檔案進行破壞的動作。



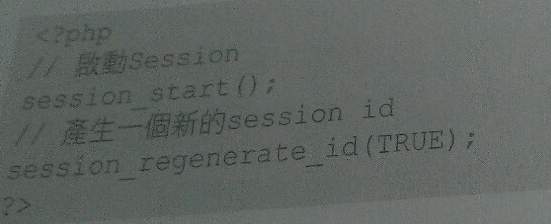
攻擊的程式檔:程式最後將用戶訂閱的書數量改為2000本



防禦方法

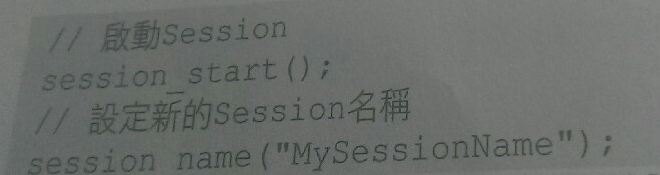
1.定期更換session id:

在PHP檔案開頭使用session\_regenerate\_id函數來產生一個新的id



1. 更換session名稱

Session預設名稱是PHPSESSID，此數值會儲存在cookie，如果駭客能取得cookie就能知道session id



1. 關閉透明化session id功能

意思就是:當瀏覽器的http請求並沒有使用cookies指定session id時，就在所有的相對連結中放入session id。

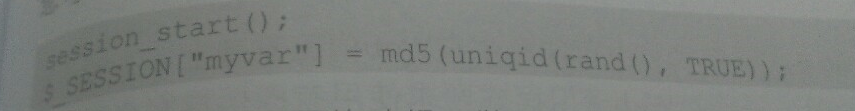
在php.ini將session.use\_trans\_sid設定為0。

1. 只從cookie檢查session id

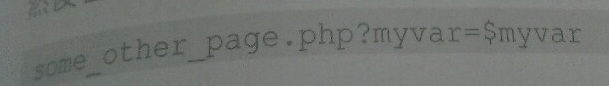
存放session id最好的地方就是cookie，所以必須將session.use.cookies定為1，另外還有一個設定是session.use\_only\_cookies設定為1，表示不使用cookie以外的方式存放session id。

1. 使用URL傳遞秘密參數

在網頁中隨機建立一個session變數



並在開啟每個網頁都將此變數值當URL參數傳遞



這樣即使駭客能使用seeion持攻擊來存取資料，但他無法得知此變數的值，因此只要檢查URL參數的值就可以確認這個網頁是否是WEB應用程式自己呼叫的。

**8.目錄遊走攻擊**

駭客能在Web應用程式所在的跟目錄以外的資料夾上，任意存取被限制的資料夾，也稱path traversal攻擊。

為了避免使用者存取伺服器尚未經許可的檔案，伺服器提供兩個安全機制:

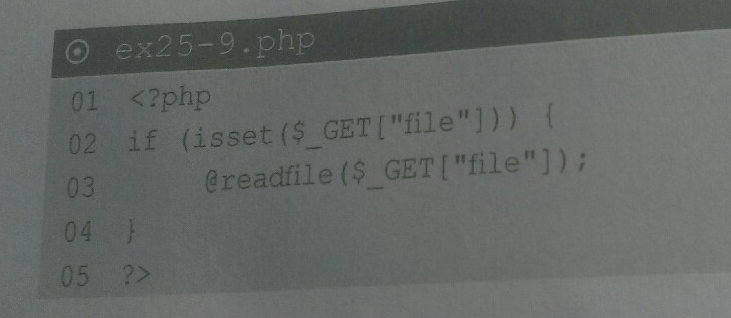
1. 根目錄
2. 檔案存取權限

伺服器限制使用者只能存取web應用程式所在根目錄以及根目錄底下的次目錄中的檔案，更進一步限制使用者只能存取根目錄中特定檔案，則由網站管理員使用檔案存取權限決定使用者可以檢視、執行、或存取什麼檔案。

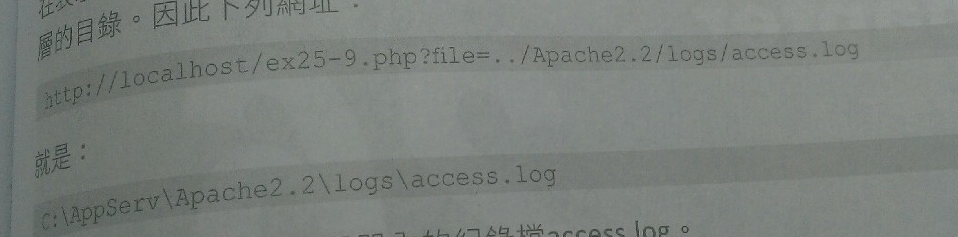
攻擊實例

範例檔位於dirTra資料夾

下圖為PHP檔案讀取一個檔案名稱，並由readfile函數輸出檔案內容。

****

如果駭客使用下列URL



這表示要開啟伺服器內的紀錄檔，駭客利用目錄遊走攻擊能夠隨便的執行網站內私密檔案。

防禦方法

1. 不要使用使用者提供的檔名
2. 檢查使用者輸入的檔名是否有..，來指定可以開啟檔案的目錄
3. 在php.ini檔案中設定，open\_basedir來指定可以開啟的目錄
4. 在php.ini檔案中設定allow\_url\_fopen為off，讓web應用程式不能開啟遠端檔案
5. 使用realpath與basename函數，處理使用者輸入的檔名。

Realpath:傳回絕對路徑

Basename:傳回不包含路徑的檔案名稱