Apache

- **i** 編輯模式 **esc** 指令模式 **:w(write)q(quit)**儲存離開
- G => 移動到這個檔案的最後一列、gg => 移動到這個檔案的第一列
- more 軟體內常用指令：
  - **/關鍵字** 可以查詢 關鍵字 **空白鍵**=> 可以向下 向後翻頁
  - **q**=>離開不再查詢文件
- less 軟體內常用指令：
  - **/關鍵字 空白鍵同上、[pageup]**向前 向上翻頁、**[pagedown]**向下 向後翻頁、**gg G q 同上。**
- rm -f /etc/httpd/conf.d/welcome.conf -f: force(強制), ignore nonexistent files(忽略不存在的文件), never prompt(不提示)。
- #server's response header 設置伺服器的回應標頭中顯示的版本資訊、操作系統資訊的程度。
- ServerTokens Prod[uctOnly]/Major/Minor/Min[imal]/OS/Full
  - 顯示 Apache 產品名稱 Server sends (e.g.): Server: Apache
  - 顯示 Apache 的主要版本號 Server: Apache/2
  - Apache 的主要版本號和次要版本號 Server: Apache/2.0
  - Server: Apache/2.0.41
  - 顯示操作系統的資訊 Server: Apache/2.0.41 (Unix)
  - 顯示完整的版本資部 Apache/2.0.41 (Unix) PHP/4.2.2 MyMod/1.2
- KeepAlive On
  - Boolean 值，為 On 時，客戶端與伺服器之間的連接可以在一個 HTTP 請求和回應之後保持打開狀態，以便在後續的請求和回應中重複使用它
  - 承載多個 HTTP 請求和回應、減少 TCP/IP 建立的次數。
- 檢測連接埠的服務/版本資訊
  - nmap sV //scan version 掃描版本

資料庫環境

- **su** 以 root 權限做一些更改檔案等小動作。**su –** 進行複雜的系統管理(path mail)
- **sudo su** 用來取得 root 或是其他帳號的權限，輸入自己的密碼
- **mysql -u root -p** 資料庫系統登入 -u 為使用者登入 使用者名稱 遮蔽密碼效果
- **create database securitytest**(建立 database) / **show databases;** (顯示 All databases)
- **show tables;** 確認所建立的資料表是否成功 / **describe studata;** studata 資料表結構(Field type key..)
- **drop tables hellolinux** 刪除 hellolinux 的資料表 / **select * from studata ;** 顯示資料表中所有資料
- **Mysql/MariaDB 加解密函數**
  - **雙向：能用 key 加密後也用 key 解回明文的，DES、AES。**
  - **單向：只能加密不能解密。Md5、sha。**
- **雙向加密（加密程度較弱）**
  - **ENCODE(str , pass_str), DECODE(crpty_str , pass_str)**
    - 明文（str）或密文(crpty_str）的字串
    - 作為加密或解密基礎的金鑰(其中 pass_str 就是 key 值)
    - BLOB 儲存任意長度的二進制資料(str/ crpty_str/pass_str 等都必須是二進制 BLOB)
    - encode('密碼‘金鑰’)
    - update Userinfo set user_PW =encode('qwe456','test1') where user_SN =1;
    - insert into Userinfo (user_ID,user_PW,user_Name) values ('tgh963da',encode('rtg964da','test1'),'蕭勁藤’);
    - decode(欄位名稱,’金鑰’)

- select * , decode(user_PW,'test1') as decode from Userinfo ;

- **DES_ENCRYPT( )、DES_DECRYPT()**
  - 資料加密標準(Data Encryption Standard,DES)
  - 對稱密鑰加密區塊碼演算法、非安全的加密因使用的 56 位元金鑰過短
  - des_encrypt('密碼',金鑰')
  - insert into Userinfo user_ID,user_PW,user_Namevalues('96weq85',des_encrypt('63gwes84','test3'),'王大福');
- **AES_ENCRYPT(str,key_str), AES_DECRYPT(crypt_str,key_str)**
  - 進階加密標準(Advanced Encryption Standard,AES)
  - 替代原先的 DES、需要 Linux、AES_ENCRYPT 加密結果最好以 BLOB 類型儲存

- 單向加密
  - **ENCRYPT(,)**
    - 第一個引數為需要加密的資料(CHAR、VARCHAR、BLOB)、第二個引數則為加密所需的金鑰(Char or BLOB)、windows 上不支援
    - 使用 UNIX crypt() 系統加密字串
  - **PASSWORD()**
    - 創建一個加密的密碼字串，MySQL 的安全獲認證系統。加密過程不可逆，和 unix 密碼加密過程使用不同演算法。
    - count 比對加密後的密碼
- 隱碼攻擊
  - SELECT * FROM user WHERE email = '$email '
    AND password = '$password'
  - 將$emali 改為' or 1 = 1-- ，$password 改為隨意值
  - SELECT * FROM user WHERE email = ' ' or 1 = 1--'
    AND password = '(隨意值) ' 等於執行 SELECT * FROM user
  - SQL 中兩個單引號等於一個單引號 ，前方單引號是為了取消 email 的輸入格
  - or 1 = 1 會讓 where 的值等於 1(ture)、-- 透過註解掉後面的條件(AND)來只執行 ' or 1 = 1 的條件。

網頁防火牆

- **靜態網頁**
  - 無法跟使用者有所互動
- **動態網頁**
  - 資訊庫功能結合
  - 使用者輸入參數(帳密)
- **OWASP TOP 10**
  - **Injection(注入攻擊)-1**
    - 未對使用者輸入的參數值進行適當的驗證或是過濾
    - 可利用惡意的輸入值(sql 指令、script 碼)
    - 對系統而言輸入惡意字元依然可以正常的通訊交換
  - **Injection(注入攻擊)-2 、SQL injection(又稱為隱碼攻擊) 、command injection**
    - SQL injection 是造成資料庫的危害
    - 可以利用特殊字元或 SQL 語句，來繞開應用程式的查詢邏輯，取得資料庫的資料。
    - select * from account where account='' or 1=1 /*' and password='' ，在/*後面的字串通通沒有執行，判斷式 1=1 永遠成立。

- **Broken authentication and session management(無效身分認證及連線管理漏洞)**
  - Session 並未設定適當的離線時間、cookies 檔案未適當的加密處理、密碼強度不足
  - 應用程式的身分認證和連線管理機制不完善，攻擊者可以利用這些漏洞，透過猜測密碼、竊取 Session ID 等
- **Cross-site scripting (XSS，跨網站腳本攻擊)-1**
  - 和 SQL injection 發生的原因相同，都因程式開發者未嚴格限制，使用者輸入與未過濾特殊字串，讓惡意的 Script 在使用的瀏覽器上執行。
  - XSS 攻擊是造成使用者瀏覽網站上的危害
- **Cross-site scripting (XSS，跨網站腳本攻擊)-2**
  - <ifram name= 'test1' width=0 height=0 src=" 惡意攻擊所在位置" ></iframe>
  - <script src src="http://3b3.org/c.js></script>讓瀏覽器執行，惡意程式碼
- **Insecure Direct Object References(不安全的物件參考)**
  - 未對使用者所輸入之參數值進行驗證進而造成的網路安全漏洞
  - GET 傳值 https://www.cdewebsite.com/user?account=1234
- **Security Misconfiguration (不安全的安全組態設定)**
  - 某些網站、裝置、作業系統等的預設帳戶、密碼、路徑，使用者在安裝時忽略安裝、使用手冊的提醒未將預設值重新設定。
- **Sensitive Data Exposure(敏感資訊外洩)**
  - 敏感性資料沒有透過適當的加密保護
  - http 通訊協定傳送資料 在資料傳輸時未使用加密(明碼)方式傳送，被攻擊者利用 Sniffer(竊聽)方式來取得傳輸的資料
- **Missing Function Level Access Control(不安全的功能控管)**
  - 網頁程式具有超連結的特性，讓攻擊者想要控制整個系統的方式就更多元。(讓進入點只有少數)
- **Cross Site Request Forgery(跨網站冒名請求, CSRF)**
  - 廣義的跨網站攻擊(X.S.S)通常為使用者登入的情況下
  - 點擊按鈕就會觸發程式自動登出你所使用的帳號、洩漏密碼、個人資料或金錢等
- **using Known vulnerable components(使用有已知安全漏洞的模組或元件)**
  - 元件具有某些安全漏洞，開發出來的軟體也將繼承它的漏洞
- **Unvalidated Redirects and Forwards(未驗證的網頁重新導向)**
  - 轉址漏洞、利用觸發程序將使用者重新導向其他網站或前往其他頁面，下載到惡意攻擊的程式碼。
- **Ps aux | grep httpd**
  - Ps 將某個時間點的程序運作情況擷取 a: 顯示其他用戶啟動的所有 process u:查看系統中屬於自己的 process x:查看這個行程的用戶和啟動時間 | 管線指令 grep 正規表示法進行全域尋找
    以及列印
  - 它將會列出正在運行的所有進程，並過濾包含 httpd 關鍵字的進程

- **Nmap sV 127.0.0.1 (找到 Microsoft IIS/5.0")**
  - 利用 syn 掃描來探測被探測主機所開啟埠資訊
  - 檢測連接埠的服務/版本資訊

網路安全概論與實務 CH2

- 駭客決定攻擊目標前，第一步驟會先行探測目標主機的 OS。由於每一家 OS 對於封包處理方式不同，探測軟體即可透過此特性來辨識 OS。此種方式就像人類指紋一樣，又稱(作業系統指紋)
- **作業系統指紋**
  - 主動式
    - 傳送特別的 TCP、UDP、ICMP 封包至被探測主機
    - 根據被探測主機回傳的封包特徵，來辨識出該主機的作業系統
    - 工具:Namp
  - 主動式作業系統的辨識原理
    - FIN 封包探測或 XMAS 封包探測
      - 掃描軟體送出一個 FIN 封包(未設置 ACK or SYN 標記位元的資料包)至主機上，在正常的 TCP/IP 規範下，對此類封包式不予理會。
        但是某些 OS 上(Window、cisco 等)，會回應 Reset 封包，因此可被用來辨識 OS。
        也由於此種封包在封包表頭上長相像聖誕樹，所以又稱 XMAS(Christmas)的封包探測。
        Port is open 意味不會回傳任何訊息(no response)、Port is closed 意味回傳 RST 封包
    - Bogus （偽造）封包探測
      - 掃描軟體在 SYN 封包的 HEADER 上設置一個未定義的 TCP FLAG 值，正常下對此封包是不予理會但在某些 OS 上遇到此封包(SYN+Bogus)時，會回傳 RST 封包來重置連線，即可利用此特徵來辨識 OS。
    - TCP initial windows (TCP 初始化視窗)探測
    - ICMP TOS (Type Of Service)判別
      - Internet Control Message Protocol （網際網路訊息控制協定），錯誤偵測與回報的機制，來檢驗網路的連線狀態與連線的正確性。
      - 也可探測 OS 的種類，利用 ping 程式送出一個 icmp echo 請求至要探測主機上，再由對方回的 echo reply 封包中的 TTL(Time To Live)來判斷 OS。Windows 126、Linux 64
  - 被動式
    - 利用 sniffer （嗅探）技術來監看往來封包
    - 藉由往來封包的特徵來辨識出作業系統
    - 工具: p0f
- 探測主機的通訊(Port) **S**ources 來源端 **D**estination 目的端
  - 全連接掃描
    - 1.**S** 發送 SYN 封包 2.**D** 回覆 SYN+ACK 封包 3.**S** 回覆 ACK 封包 連線建立(稱為三方交握(Three Handshake))
    - 一旦掃描軟體被探測主機 完成三向交握並與主機完成連線，代表 Port 為開啟，反之亦然。此種掃描為全連接掃描。 因屬於完成的連線，所以代表此 scan 會被 firewall or IDS 所記錄。
  - 半連接探測技術
    - 由於使用全連接方式掃描，容易被 Firewall or IDS 所記錄，所以會使用半連接。
    - SYN 封包掃描
      - 在對方主機回傳 SYN/ACK 封包時候，代表 PORT 為開啟，這時發送 RST 封包切斷連接。
    - SYN/ACK 封包掃描
      - 是利用繞過 Three Handshake 第一步(SYN)，直接使用 SYN/ACK 封包至目的主機上。
      - 如果 Port 為開啟會發現不是合法封包，進而丟棄(no response)。若 Port 為關閉則會回傳 RST 封包(RST)。

- FIN 封包掃描 同 SYN/ACK 技術判別方法

- **scanlogd**
  - 埠掃描偵測軟體來建立埠掃描偵測系統，一旦發現惡意埠掃描便立即將相關資訊寫入 var /log/

- #cat /etc/passwd
  存儲在 Unix 和類 Unix 系統上的用戶帳號和密碼的檔案
  用戶名稱、用戶 ID、群組 ID、登錄 shell、家目錄
- #cat /etc/passwd|md5sum
- 雜湊值 固定長度的字元串，示檔案的唯一數字指紋
- 計算它的 md5 雜湊值，驗證該檔案是否正確且未被篡改

- #ifconfig 顯示當前系統上可用的網路介面的信息，包括 IP 地址、子網掩碼、MAC 地址

- Ping
- 測試網路品質
- Linux 的 Ping 数据包是 64bytes 的而 Windows 的是 32byte

- Ping -t 一直送資料測試
- Ping -n,-l,-l,-w 數字
  - -n : 做 10 次測試
  - -l 64 : 64 位元組測試
  - -l 10 : 僅能通過 10 個網站 不指定為 255
  - -w 1000 : 等待對方網頁回應為 1000ms