Web-based Authentication Wifi Access Point

無線網路實驗一 報告 608410117 沈濃翔

Wifi-auth

Github: https://github.com/Yuki23329626/wifi-auth/
所有用到的 config files 和 script 都在 github 上了
雖然後面也有貼 code 跟說明，不過比較亂，可以改到 github 上看

後面會針對每一個用到的檔案進行註解說明，

先講基本的架設操作，如下：

OS：ubuntu 16.04
需要兩張網卡，一張給內網、一張給外網

學校筆電因為裝了 nvidia 顯卡，ubuntu 16.x 會卡開機畫面
解決方法：https://itsfoss.com/fix-ubuntu-freezing/
1. 進到開機 USB 選單，press 'e' 進入 grub 畫面
2. 編輯開頭是 linux 的那一行命令，最後面 "---" 改成 "nomodeset" 這個字串

個人建議不要用 ubuntu 18.x 的版本來練習，各種神奇的 features

1. 一個指令搞定所有安裝 + iptables 設定

總之就是先執行腳本,在 bash 輸入以下指令：

```
sudo sh wifi-ap.sh
```

關於 shell script 內用到的檔案，
都需要事先改好 network interface 的名稱
在我的電腦上 wlp2s0 是在內網的 interface
而 wlxf48ceb9ba387 是連接外網的 interface
具體要怎麼查詢，請使用下列 command:

```
ifconfig -a
```

ip address 為 10. 或是 192. 開頭的通常會作為內網分配的 IP 使用
lo 是 localhost 的介面
剩下的就是可以連到 internet 的介面與實體線路的介面
實體線路沒插線設定不會有 IPv4 的 address 應該不難分辨

可能需要更改 interface ID 的檔案：

```
- hostapd.conf
- interfaces
- isc-dhcp-server
- wifi-ap.sh
```

以上檔案都需要設定內網網卡 ID，只有 wifi-ap.sh 需要再設定外網網卡 ID
wlp2s0 是我內網網卡的 ID，新電腦可能都一樣，舊電腦可能會叫做 wlan0 之類的

2. apache 權限設定

apache 部份相關檔案需要手動設定

```
sudo visudo
```

在檔案裡加上這一行:

```
www-data  ALL=(ALL)NOPASSWD:/sbin/iptables
```

這個動作的目的是要讓 apache 在 linux 系統裡的身分(www-data)
有權限去執行 auth.cgi 裡面的 system() iptables 設定

3. 匯入 mysql 資料

匯入資料的 SQL 請參照 repository 內的 'db_init_sql.txt' 檔案內容
基本上就是進入 mysql 的 command line 之後直接複製貼上就行了

4. mysql 帳密

進入 mysql cmd，使用'root'這個帳號，並且要輸入密碼，我是用'secret'做密碼
若需要改帳號密碼，則相關的 auth.cpp 等檔案的帳密也需要修改

```
mysql -u root -p
```

File name: wifi-ap.sh

```
#!/bin/bash
# Progra:
# This program will set up a wifi access point with a web-
based authentication and also set up the iptables.
# History:
# 2019/11/8 nxshen add several comments
PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin:~/bin
export PATH
echo "\nHello there, it's a shell script for establishing a wifi access point~\n"

# 設定網卡 ID(以下腳本會用到的變數)
LAN_INTERFACE=wlp2s0
WAN_INTERFACE=wlxd037453d9c6a

echo "\n-- checking for necessary packages --\n"

# 安裝 apache2、mysql-server(也可以直接安裝 lamp-server^)、
# 安裝 dhcp-server(動態分配內網 IP 的 server)、dnsmasq(DNS server)

# PKG_OK=$(dpkg-query -W --showformat='${Status}\n' lamp-
server^|grep "install ok installed")
# echo Checking for lamp-server^: $PKG_OK
# if [ "" = "$PKG_OK" ]
# then
#   echo "Have not installed. Start installing..."
#   sudo apt install lamp-server^
# fi
PKG_OK=$(dpkg-query -W --showformat='${Status}\n' apache2|grep "install ok installed")
echo Checking for apache2: $PKG_OK
if [ "" = "$PKG_OK" ]
then
  echo "Have not installed. Start installing..."
  sudo apt install -y apache2
fi
PKG_OK=$(dpkg-query -W --showformat='${Status}\n' mysql-
server|grep "install ok installed")
echo Checking for mysql-server: $PKG_OK
if [ "" = "$PKG_OK" ]
then
  echo "Have not installed. Start installing..."
```

```
  sudo apt install -y mysql-server
fi
PKG_OK=$(dpkg-query -W --showformat='${Status}\n' isc-dhcp-
server|grep "install ok installed")
echo Checking for isc-dhcp-server: $PKG_OK
if [ "" = "$PKG_OK" ]
then
  echo "Have not installed. Start installing..."
  sudo apt install -y isc-dhcp-server
fi
#PKG_OK=$(dpkg-query -W --showformat='${Status}\n' dnsmasq|grep "install ok installed")
#echo Checking for dnsmasq: $PKG_OK
#if [ "" = "$PKG_OK" ]
#then
#  echo "Have not installed. Start installing..."
#  sudo apt install dnsmasq
#fi
PKG_OK=$(dpkg-query -W --showformat='${Status}\n' libmysql++-
dev|grep "install ok installed")
echo Checking for libmysql++-dev: $PKG_OK
if [ "" = "$PKG_OK" ]
then
  echo "Have not installed. Start installing..."
  sudo apt install -y libmysql++-dev
fi
PKG_OK=$(dpkg-query -W --showformat='${Status}\n' hostapd|grep "install ok installed")
echo Checking for hostapd: $PKG_OK
if [ "" = "$PKG_OK" ]
then
  echo "Have not installed. Start installing..."
  sudo apt install -y hostapd
fi

echo "\n-- start copying files --\n"

# 把 config files 直接放到他們該在的地方，記得修改各自 config file 內的網卡名稱設定
cp auth.cpp /usr/lib/cgi-bin/
cp auth.cgi /usr/lib/cgi-bin/
cp makefile /usr/lib/cgi-bin/
# 要先安裝完成 mysql 用的 library "libmysql++-dev" 才能成功編譯，auth.cpp 會用到 "libmysql++-
dev"
make
```

```
sudo cp envvars /etc/apache2/
sudo cp dhcpd.conf /etc/dhcp/
sudo cp hostapd.conf /etc/hostapd/
sudo cp bookmarks.html /home/
sudo cp setIptables /home/
sudo cp index.html /var/www/html/
sudo cp isc-dhcp-server /etc/default/
sudo cp dhcpd.conf /etc/dhcp/
sudo cp interfaces /etc/network/
#sudo cp dnsmasq.conf /etc/
sudo cp NetworkManager.conf /etc/NetworkManager/

echo "\n-- start and enable services --\n"

# 啟動該啟動的服務們並且設為開機啟動
sudo echo "1" > /proc/sys/net/ipv4/ip_forward
sudo ifconfig $LAN_INTERFACE 10.10.0.1/24 up
systemctl start apache2.service
systemctl enable apache2.service
systemctl start isc-dhcp-server.service
systemctl enable isc-dhcp-server.service
systemctl start mysql.service
systemctl enable mysql.service
#systemctl start dnsmasq.service
#systemctl enable dnsmasq.service
sudo ufw allow  67/udp
sudo ufw reload
#sudo systemctl restart networking
sudo service network-manager stop
sudo service network-manager start

#sudo /etc/init.d/dnsmasq restart

# 允許 apache 啟用 cgi 的 module，要 restart 才會生效
sudo a2enmod cgi
sudo service apache2 restart


# 驗證網頁是: 10.10.0.1/index.html，應該也可以設定 /etc/hosts 來給他一個名稱
# 以下六行清空所有 iptables 的規則
iptables -Z
iptables -F
iptables -X
```

```
iptables -t nat -Z
iptables -t nat -F
iptables -t nat -X

# 把所有經過 filter forward chain 目標是 $WAN_INTERFACE 這個 interface 的封包全部丟掉
iptables -A FORWARD -o $WAN_INTERFACE -j REJECT


# 允許 NAT 上的 IP 可以轉換成外部 IP(規則:MASQUERADE)，與外網溝通
iptables --table nat --append POSTROUTING --out-interface $WAN_INTERFACE -j MASQUERADE

# 啟用 hostapd 服務
hostapd /etc/hostapd/hostapd.conf

exit 0
```

File name: /usr/lib/cgi-bin/auth.cpp

```
#include <iostream>
#include <vector>
#include <string>
#include <stdlib.h>
#include <mysql.h>
#include <regex>
#include <map>

using namespace std;

// 把網址裡面的兩個變數傳進來，拆成 username 跟 password 放進 map 裡，回傳 map<string, string>
map<string, string> Parse(const string& qstr){
  map<string, string> mapUser;
  // 例如網址後面是 key1=value1&key2=value2 會被拆成 key1, value1 一組， key2, value2 一組
  regex pattern("([\\w+%]+)=([^&]*)");
  auto words_begin = sregex_iterator(qstr.begin(), qstr.end(), pattern);
  auto words_end = sregex_iterator();

  for(sregex_iterator i = words_begin; i != words_end; i++){
    string name = (*i)[1].str();
    string password = (*i)[2].str();
    mapUser[name] = password;
  }
```

```
  return mapUser;
}


int main()
{
    // 可以呼叫的環境變數
    string strNames[]={
      "DOCUMENT_ROOT",
      "GATEWAY_INTERFACE",
      "HTTP_HOST",
      "REMOTE_ADDR",
      "REMOTE_PORT",
      "REQUEST_METHOD",
      "REQUEST_URI",
      "SCRIPT_FILENAME",
      "SERVER_ADDR",
      "SERVER_NAME",
      "SERVER_PORT",
      "SERVER_PROTOCOL",
      "SERVER_SOFTWARE",
      "QUERY_STRING",
      "HTTP_COOKIE"
    };


    vector<string> varNames(strNames, strNames+15);

    cout << "Content-type:text/html\r\n\r\n";
    cout << "<html>";
    cout << "<head>";
    cout << "<title>Envrionment Variables</title>";
    cout << "</head>";
    cout << "<body>";
    cout << "<table border = \"1\" cellspacing = \"0\">";

    // 印出環境變數比較好觀察
    for (int i = 0; i < varNames.size(); ++i)
    {
        cout << "<tr><td>" << varNames[i] << "</td><td>";
        const char *value = getenv(varNames[i].c_str());
        if (value != NULL) {
            cout << value;
```

```
        } else {
            cout << "Not exist";
        }
        cout << "</td></tr>";
    }
    cout << "</table>";
    cout << "</body>";
    cout << "</html>";

  // mysql 連線的初始化設定
  MYSQL mysql;
  mysql_init(&mysql);
  int res;
  MYSQL_RES *result;
  MYSQL_ROW sql_row;

  if(!mysql_real_connect(&mysql, "localhost", "root", "secret", "wifi_auth", 3306, NULL, 0
)){
    cout<< "\nError connecting to database\n" << mysql_error(&mysql) <<"\n\n";
  }else{
    cout<<"MySQL database Connected!\n";

    //mysql_query(&mysql, "SET NAMES UTF8");

    string qstr = getenv(varNames[13].c_str());
    cout << "<BR>" << qstr <<"<BR>";
    map<string, string> mapUser = Parse(qstr);

    //cout <<"<BR>mapUser.first: "<< mapUser.first <<"<BR>";
    //cout <<"<BR>mapUser: "<< mapUser["name"] <<"<BR>";

    // 印出網址裡的 "user" 對應的 value
    auto iterUser = mapUser.find("user");
    if(iterUser != mapUser.end()){
      cout << "<BR>mapUser[\"user\"]: "<< iterUser->second <<"<BR>";
    }

    // 印出網址裡的 "pass" 對應的 value
    auto iterPass = mapUser.find("pass");
    if(iterPass != mapUser.end()){
      cout << "<BR>mapUser[\"password\"]: "<< iterPass->second <<"<BR>";
    }
```

```
    // SQL 語法，找出跟網址裡一樣的 user name
    mysql_query(&mysql, "use wifi_auth");
    string dbQuery = "select * from user where name=\'"+ iterUser->second +"\'";

    //string strQuery = "select * from user";
    //res = mysql_query(&mysql, "select * from user");
    res = mysql_query(&mysql, dbQuery.c_str());
    if(!res){
      result = mysql_store_result(&mysql);
      if(result){
        /*cout << "<table border = \"1\" cellspacing = \"0\">";
        while(sql_row = mysql_fetch_row(result)){
          cout << "<TR><TD>" << sql_row[1] << "</TD>";
          cout << "<TD>" << sql_row[2] << "</TD></TR>";
        }*/
        sql_row = mysql_fetch_row(result);

        // 如果找到 user name 之後，比對成功的話，就會進入下面的 block，印出成功訊息並且設
定 iptables
        if((iterPass->second) == (sql_row[2])){
          cout << "<BR>login success!<BR>";
          string strRemoteAddr(getenv(varNames[3].c_str()));
          // iptables 在 filter 的 forward chain 插入規則: 只
要 source ip 跟 destination ip 是 remote address，就通過
          string str1 = "sudo iptables -I FORWARD -s " + strRemoteAddr + " -j ACCEPT";
          string str2 = "sudo iptables -I FORWARD -d " + strRemoteAddr + " -j ACCEPT";
          cout << "<BR> str1 = " + str1 + "<BR>";
          cout << "<BR> str2 = " + str2 + "<BR>";
          const char* cmd1 = str1.c_str();
          const char* cmd2 = str2.c_str();
          int return1 = system(cmd1);
          int return2 = system(cmd2);
          // int return3 = system("echo 1");
          cout << "<BR>system(cmd1) retruns " << return1 << "<BR>";
          cout << "<BR>system(cmd2) returns " << return2 << "<BR>";
          // cout << "<BR>system(cmd3) returns " << WEXITSTATUS( return3 ) << "<BR>";
        }else{
          cout << "<BR>login failed!<BR>";
        }
      }
    }
```

```
  }

return 0;

}
```

File name: /usr/lib/cgi-bin/makefile

```
# 用來編譯 auth.cpp 程式的 makefile
# 會使用到 mysql 的 library "libmysql++-dev"，記得先安裝完成 "libmysql++-dev"
all: auth.cpp
  g++ -std=c++11 -I/usr/include/mysql auth.cpp -L/usr/lib/mysql -lmysqlclient -
o /usr/lib/cgi-bin/auth.cgi
clean:
  rm auth.cgi
```

File name: db_init.sql

```
-- 用來初始化 mysql database 並輸入資料的 SQL
SHOW DATABASE;
CREATE DATABASE wifi_auth;
USE wifi_auth;
DROP TABLE IF EXISTS `user`;
CREATE TABLE `user` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` text COLLATE utf8mb4_unicode_ci NOT NULL,
  `password` text COLLATE utf8mb4_unicode_ci NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-- 匯入使用者跟密碼到 'user' 這個 table
INSERT INTO `user` (`id`, `name`, `password`) VALUES
(1, 'user001',  '001001'),
(2, 'user002',  '002002'),
(3, 'user003',  '003003'),
(4, 'user004',  '004004'),
(5, 'user005',  '005005'),
(6, 'user006',  '006006');
```

File name: /etc/dhcp/dhcpd.conf

```
# DHCP 設定
# 如果想要幫 10.10.0.1 取名，可能需要到 /etc/hosts 新增 domain name
# option domain-name "nxshen.lan";
option domain-name-servers 10.10.0.1;

# 子網域的設定
subnet 10.10.0.0 netmask 255.255.255.0 {
        range 10.10.0.2 10.10.0.128;
        option domain-name-servers 10.10.0.1;
        option routers 10.10.0.1;
}
```

File name: hostapd.conf

```
# 記得把 interface 改成自己想要用來當內網網卡的 ID
interface=wlp2s0
driver=nl80211
ssid=test
channel=6
hw_mode=g
auth_algs=1
wpa=0
wpa_passphrase=19960415
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

File name: /var/www/html/index.html

```
<!-- 網頁驗證的登入頁面，目前需要手動輸入網址: 10.10.0.1/index.html -->
<meta http-equiv="Content-Type" content="text/html; charset=utf8">
<title>Authentication Page</title>
</head>

<body>
    <FORM METHOD="GET" ACTION="/cgi-bin/auth.cgi">
        <p align="center">
            <b>WIFI Authentication</b>
            <BR>
            username <INPUT TYPE="text" NAME="user" SIZE=20>
            <BR>
```

```
            password <INPUT TYPE="password" NAME="pass" SIZE=20>
            <BR>
        <input type="submit" value="Submit">
        </p>
    </FORM>
</BODY>
```

File name: /etc/network/interface

```
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback

# 記得把 wlp2s0 改成自己要用來當內網網卡的 ID
auto wlp2s0
iface wlp2s0 inet static
#allowing-hotplug wlp2s0
address 10.10.0.1
netmask 255.255.255.0
```

File name: /etc/default/isc-dhcp-server

```
# 記得把 INTERFACES 改成自己要用來當內網網卡的 ID
INTERFACES="wlp2s0"
```

File name: /etc/NetworkManager/NetworkManager.conf

```
[main]
plugins=ifupdown,keyfile,ofono
dns=dnsmasq

[ifupdown]
managed=false

[keyfile] # mac 記得改成內網 interface 的 mac address
unmanaged-devices=mac:b0:c0:90:c7:c6:cc
```

以上