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詳細說明，**

**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

1. **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 才能成功編譯，auth.cpp 會用到 mysql 的 library

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

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: auth.cpp**

#include <iostream>

#include <vector>

#include <string>

#include <stdlib.h>

#include <mysql.h>

#include <regex>

#include <map>

using namespace std;

map<string, string> Parse(const string& qstr){

map<string, string> mapUser;

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()

{

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\">";

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 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<<"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>";

auto iterUser = mapUser.find("user");

if(iterUser != mapUser.end()){

cout << "<BR>mapUser[\"user\"]: "<< iterUser->second <<"<BR>";

}

auto iterPass = mapUser.find("pass");

if(iterPass != mapUser.end()){

cout << "<BR>mapUser[\"password\"]: "<< iterPass->second <<"<BR>";

}

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);

if((iterPass->second) == (sql\_row[2])){

cout << "<BR>login success!<BR>";

string strRemoteAddr(getenv(varNames[3].c\_str()));

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;

}