教程

正文：（博客持续同步更新）

第一部分：准本工作

下载工具集

http://pan.baidu.com/s/1nvxtcZB

1. 为路由刷入精简版linux系统——pandorabox
2. 刷入uboot——路由第三方系统更换工具

本部分完全参考至教程：

<http://bbs.hiwifi.com/forum.php?mod=viewthread&tid=82822&highlight=uboot>

breed uboot简介：

是一款全新的bootloader，不是Uboot，也不是Uboot的改进版，暂名为Breed。它是一种普遍用于嵌入式系统中的Bootloader,Bootloader是在操作系统运行之前执行的一小段程序。它可以用来恢复路由器的固件，可以说只要刷了Breed，你的路由器基本上刷不死了。

1.下载对应极路由型号的刷机文件

极路由1 breed-ar9331-hiwifi-hc6361.rar (75.74 KB, 下载次数: 1337) （请勿在极1原厂固件里刷breed，否则必砖无疑！！切记）

极路由1S（5661a） breed-mt7628-hiwifi-hc5661a.rar (71.54 KB, 下载次数: 2665) （这个文件对应5661a，）千万对应好型号，不要下错！！！

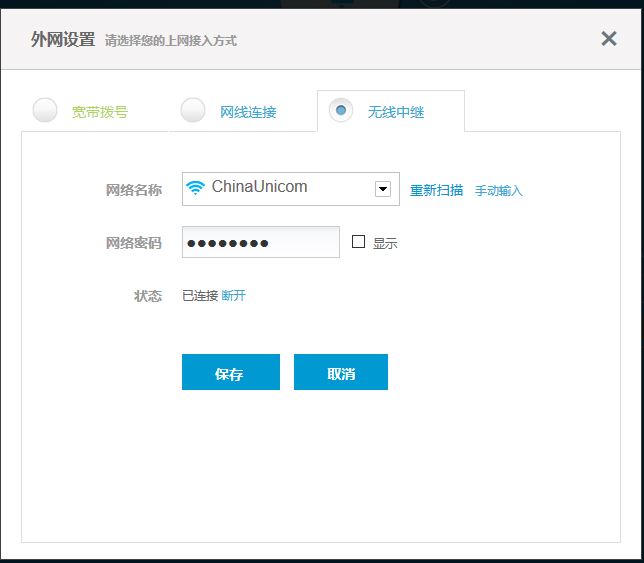
极路由2/极路由1s（5661） breed-mt7620-hiwifi-hc5761.rar (74.26 KB, 下载次数: 6527)

极路由3 breed-mt7620-hiwifi-hc5861.rar (74.22 KB, 下载次数: 2240)

2.获取极路由root权限

a，电脑连接路由lan口，登录网址192.168.1.1管理界面（管理地址刻印在路由器后的铭牌上）

进行外网设置



中继选择已有wifi，可以使用手机热点替代，保存

b，进入云平台，点击路由器信息，拉到页面最下方，点开高级设置，点击申请



服务条款页面，点击同意

通过账号绑定的手机获取口令，完成申请

之后会需要关注极路由官方微信，并且绑定自己的社区账号

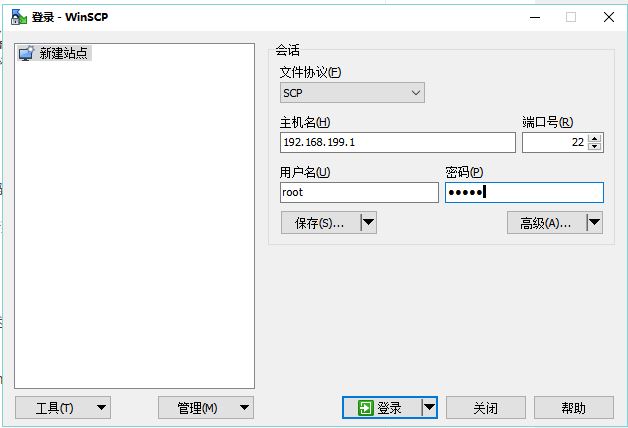
绑定后即可安装开发者模式插件，root完成

（root后会失去官方保修服务，请三思）

3.上传刷机文件

上传刷机文件需要Winscp。

下载后使用Winscp连接你的极路由，账号：root、密码 ：你的后台密码、端口：22/1022（失败了就用另一个，笔者使用1022成功）、模式：SCP

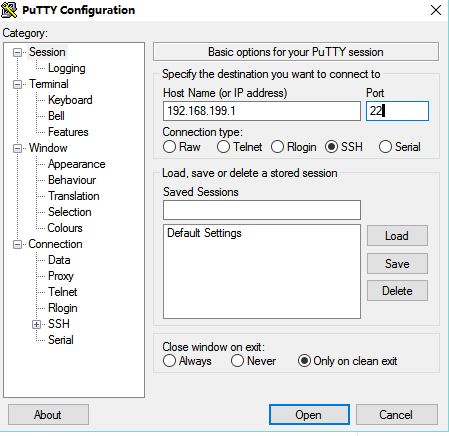


登陆成功后进入/tmp目录，将刚才下载的刷机文件上传到这个目录

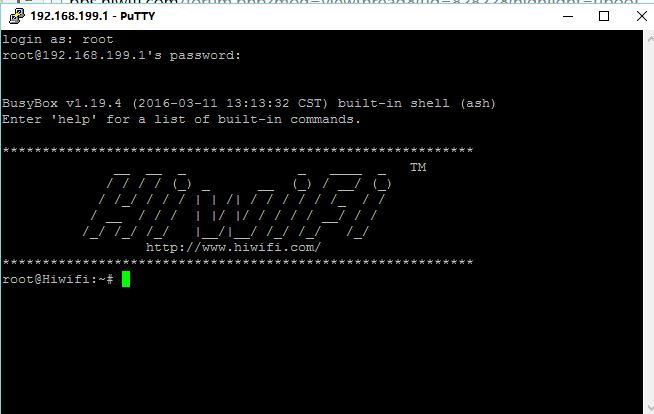
清根据机型选择，极1s出过2款cpu型号的版本，具体型号分别为5661a对应7628cpu，5661对应7620cpu，刷错GG没得救，一定看清楚

4.刷机

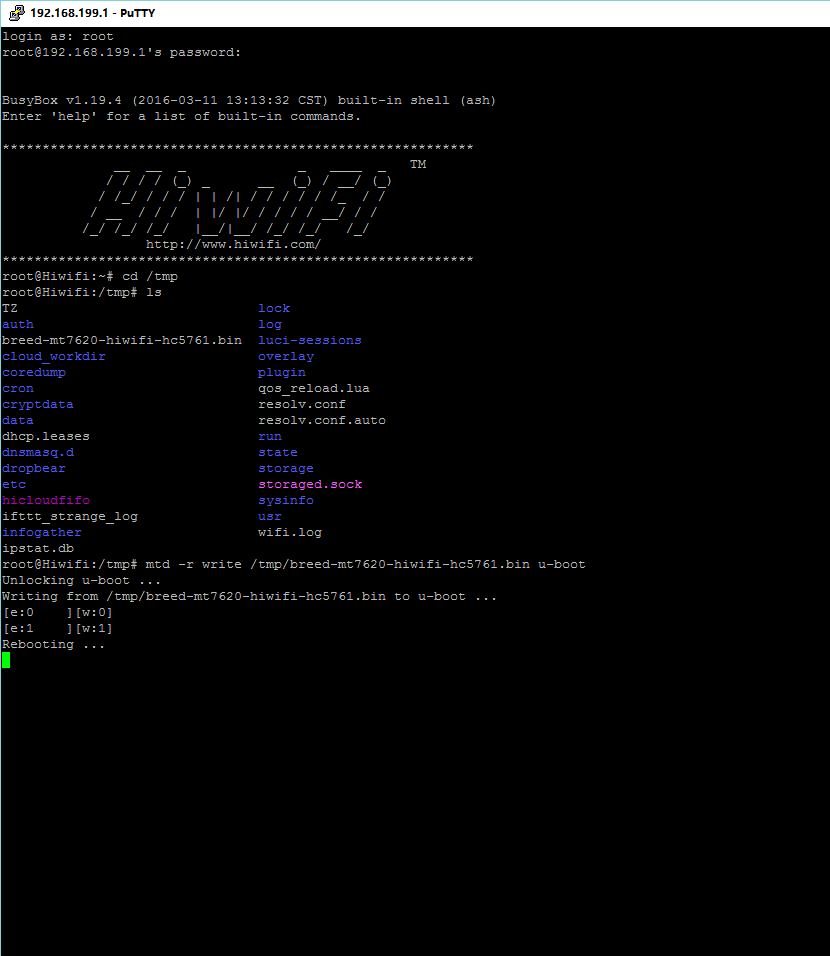
刷机需要使用putty。



下载后使用putty登入你的极路由，账号密码端口与上述相同（输入密码时不会显示，此处不必理会，只要正确输入密码即可）



登入成功后键入以下命令 mtd -r write /tmp/breed-mt7620-hiwifi-hcXXXX.bin u-boot（最后的文件名即为你下载的刷机文件名称）



显示rebooting后等待路由重启完成。

刷机结束。

1. 利用uboot刷入pandorabox系统

路由断电，按住复位键不松，重新插电（保持复位键按下），5s左右后看到路由所有指示灯闪烁，松开复位键

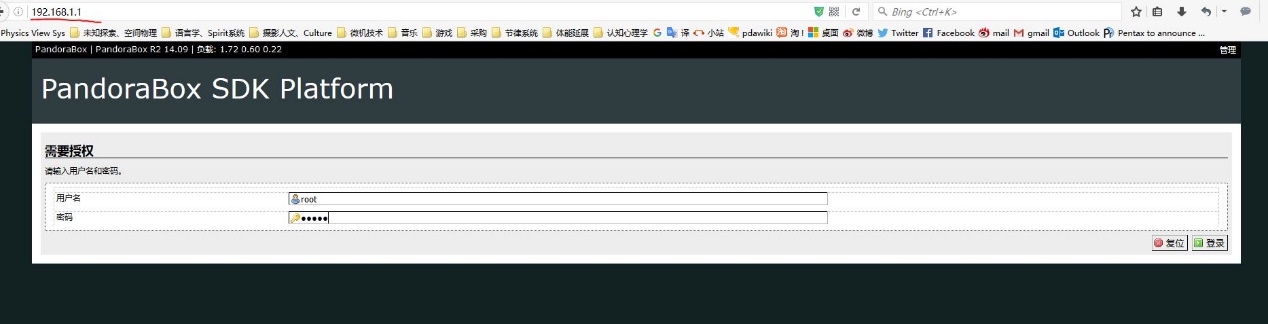
用浏览器登录192.168.1.1进入uboot刷固件系统

根据机型选择刷入的固件





刷入完毕完全重启后，重新登录192.168.1.1，出现pandorabox登录界面，恭喜刷机成功啦，默认密码是admin

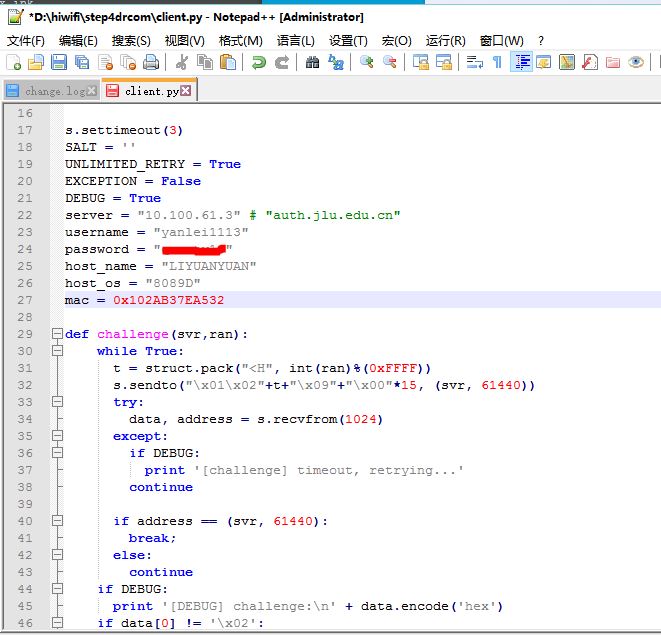


1. 查找自己的校园网帐号详情填写表单

A，使用校内网登录ip.jlu.edu.cn 自助营业厅查询自己的ip信息



B，使用下载提供的电脑软件notepad++编辑client.py（drcom文件夹中）文件，填写3个信息，username指校园网drcom登录用户名，password密码，mac为绑定的网卡地址，编辑时注意保留0x开头



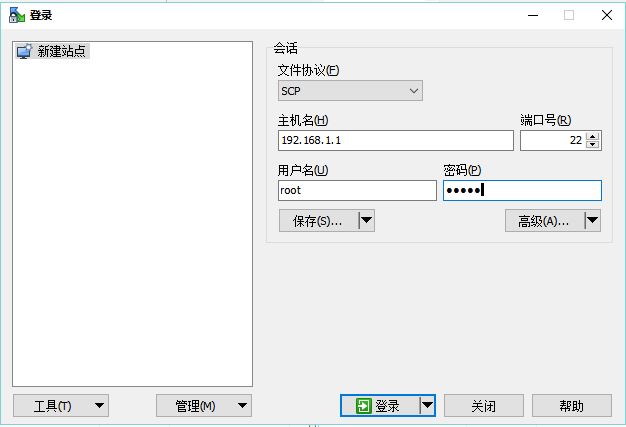
Ip等其他信息将在第四部分用到，注意保留文本

第二部分：突破校园网

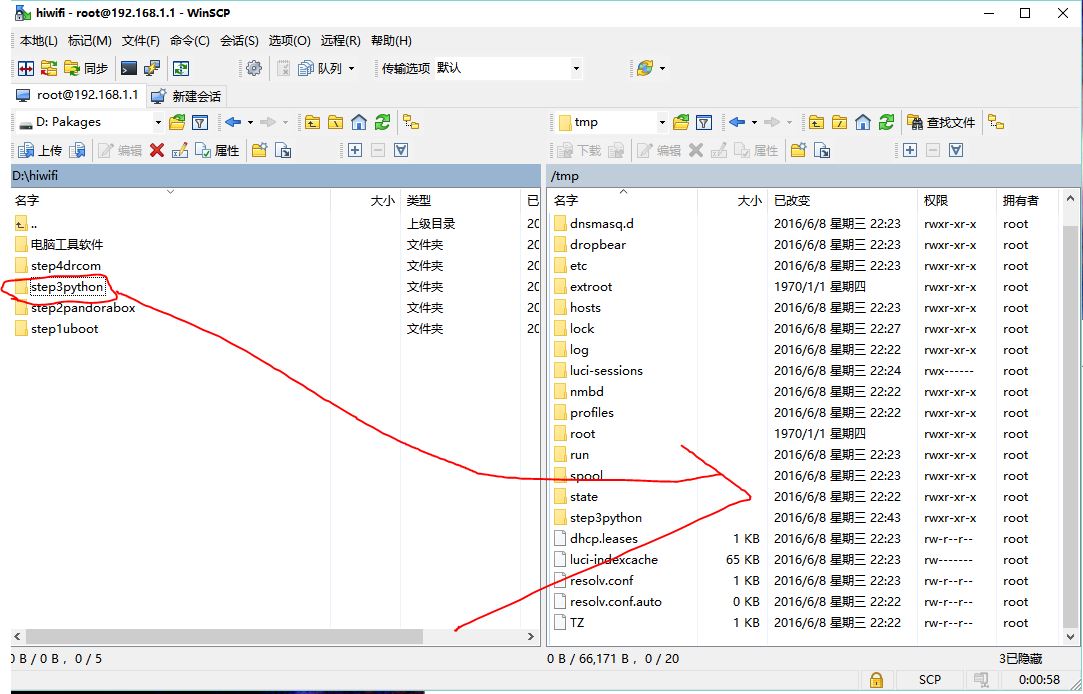
三，导入需要用到文件（安装python环境和drcom脚本）

A,导入文件

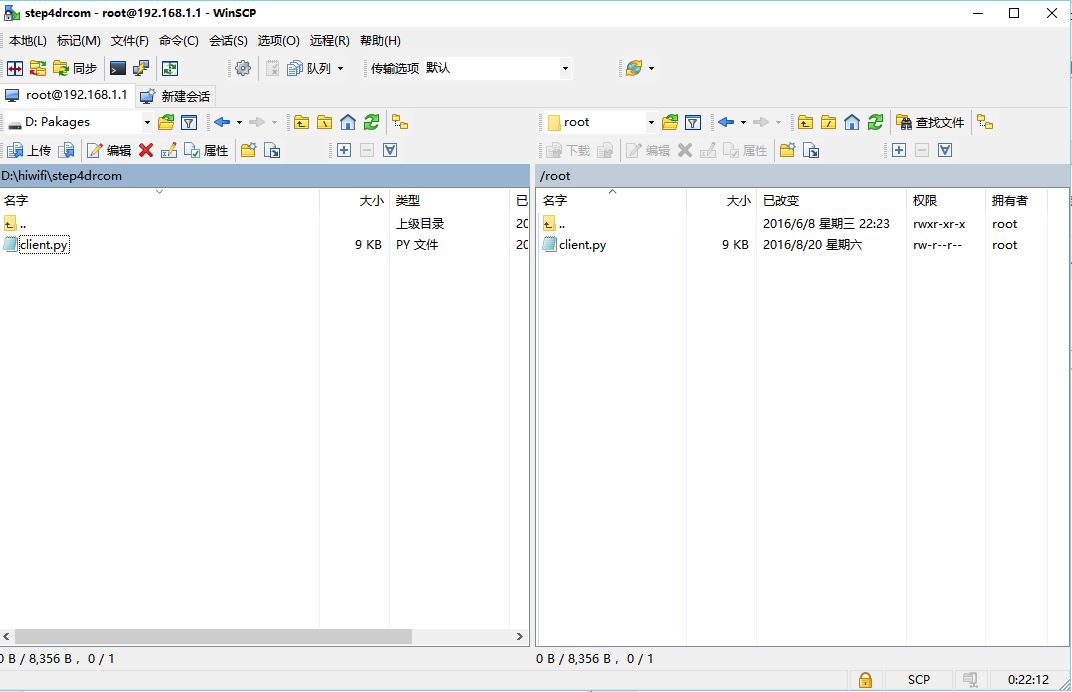
使用winscp连接pandorabox系统



拖入整个step3文件夹到tmp目录，如图

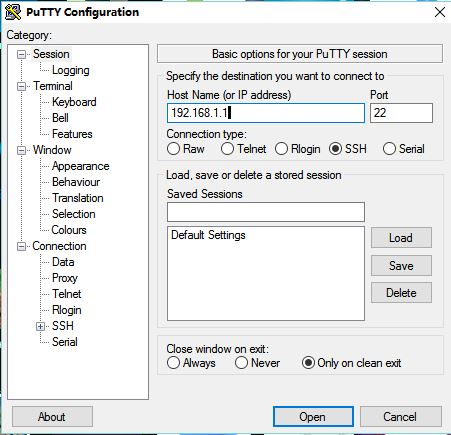


拖入根据你的信息修改，编辑保存后的client.py（drcom文件夹中）文件到/root目录，如图

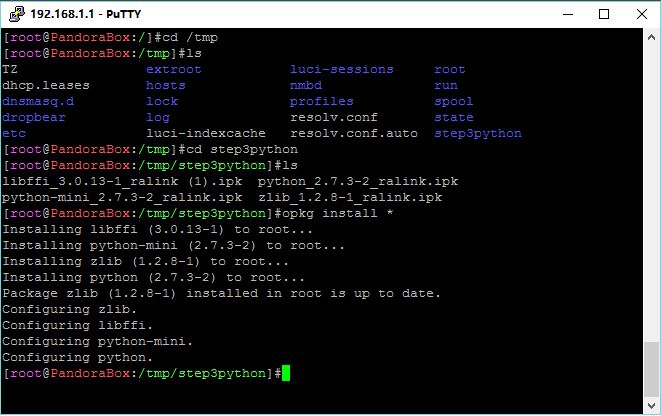


1. 安装python环境

使用putty登录pandorabox系统



使用如图指令，cd进入文件目录，ls显示当前目录下的文件，opkg安装，具体操作如图



没有报错就算成功

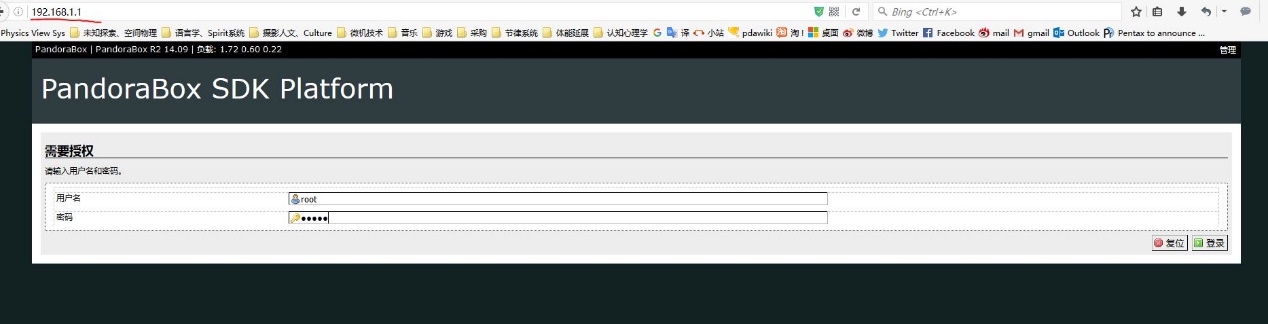
如果确定上传路径正确，也可以使用一句命令解决

opkg install /tmp/step3\*/\*l

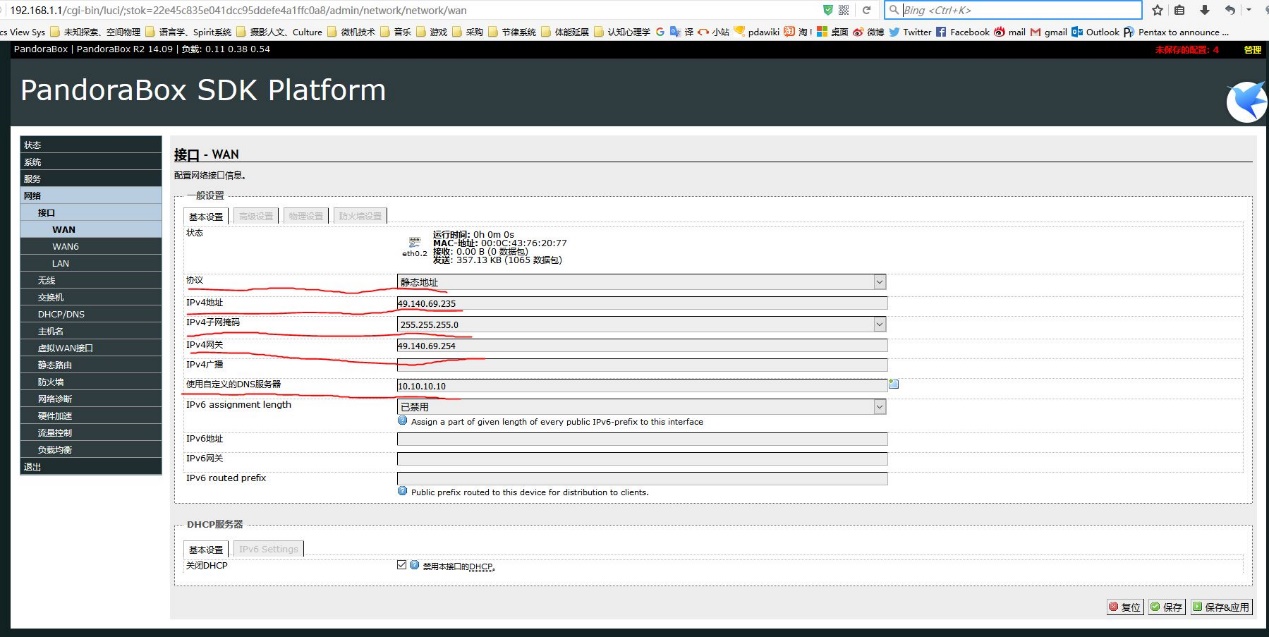
四，配置路由

使用浏览器，登录地址192.168.1.1

进入pandorabox图形管理页

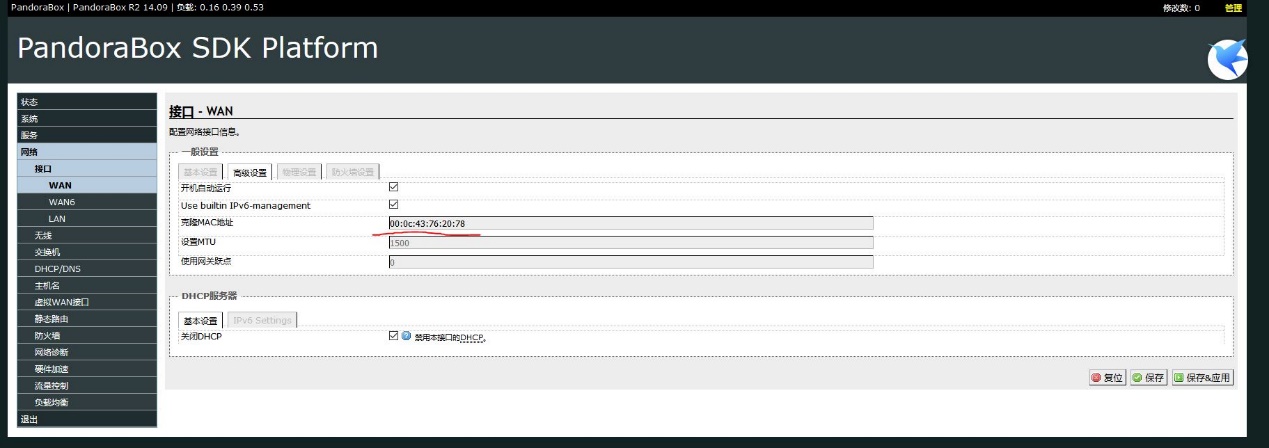


1. 设置wan接口



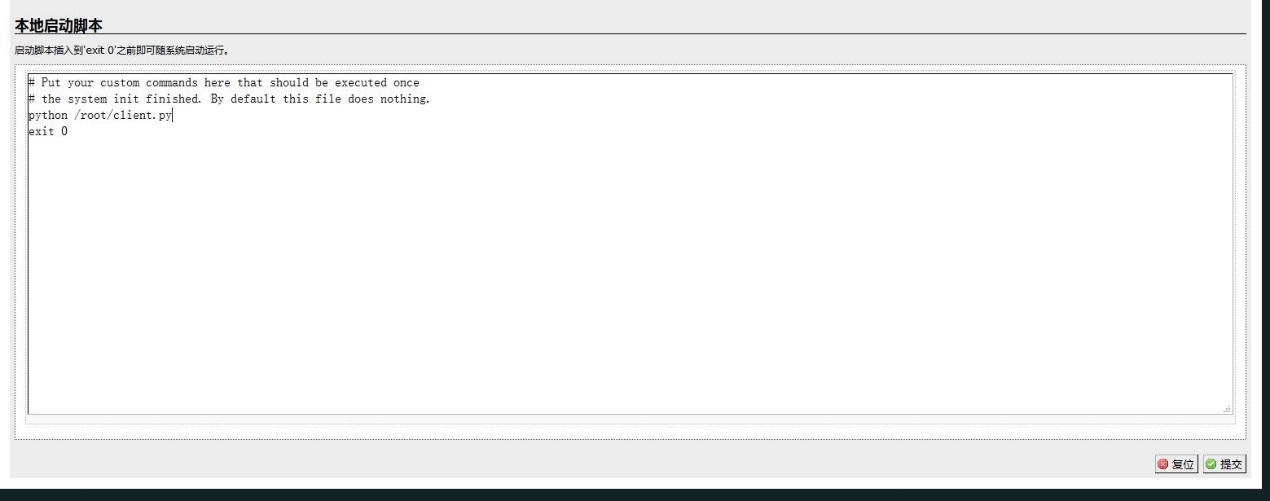
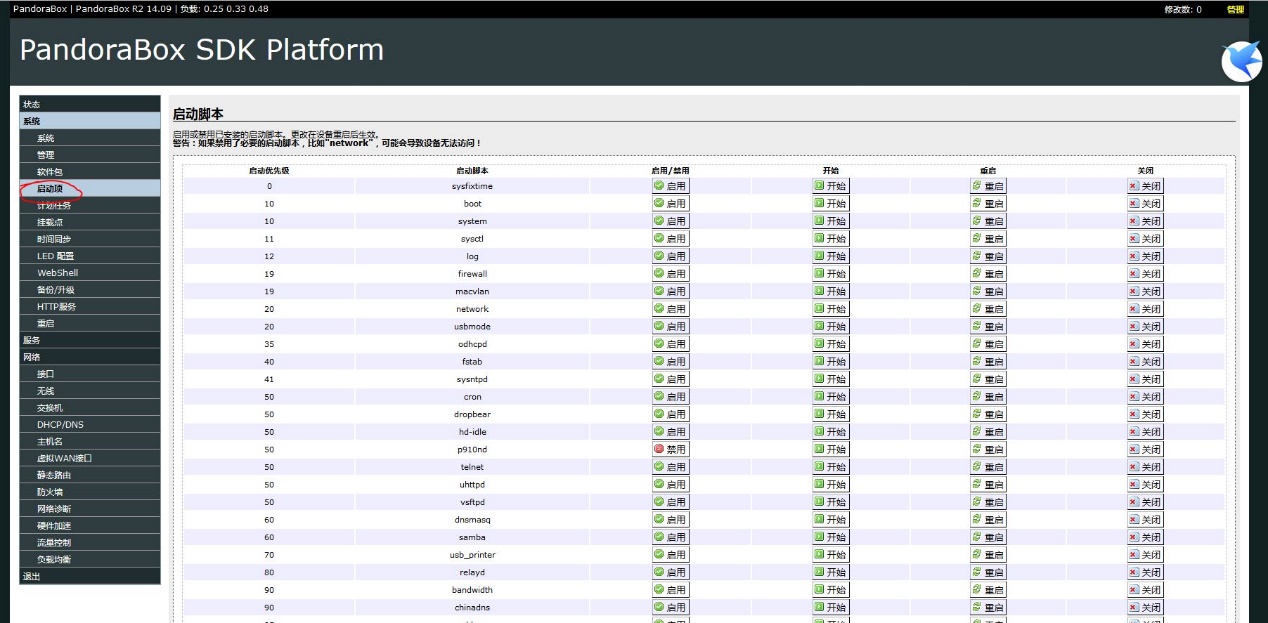
一共设置5项，先切换为静态协议，根据之前查到的信息填写，完成后点击保存

之后将副标签页切到高级设置



填写drcom绑定的mac（物理网卡）地址后保存

1. 设置启动项



加入启动命令

python /root/client.py

点击提交，保存设置

1. 设置交换机

仅仅插入总线和连接你电脑的网线，此时交换机页面应该只显示2个端口为蓝色，代表已经插入，根据相对位置可以推知哪个为总线哪个为你的电脑网线，这2个设置为一个不关联组，其他端口设为不关联组，意味着和网线同组的端口可以免拨号使用路由拨号开机上网，其他端口为正常交换机功能网线，正常设置ip后使用drcom拨号上网，互不影响，如需合并网络只需将其加入到wan口不关联组

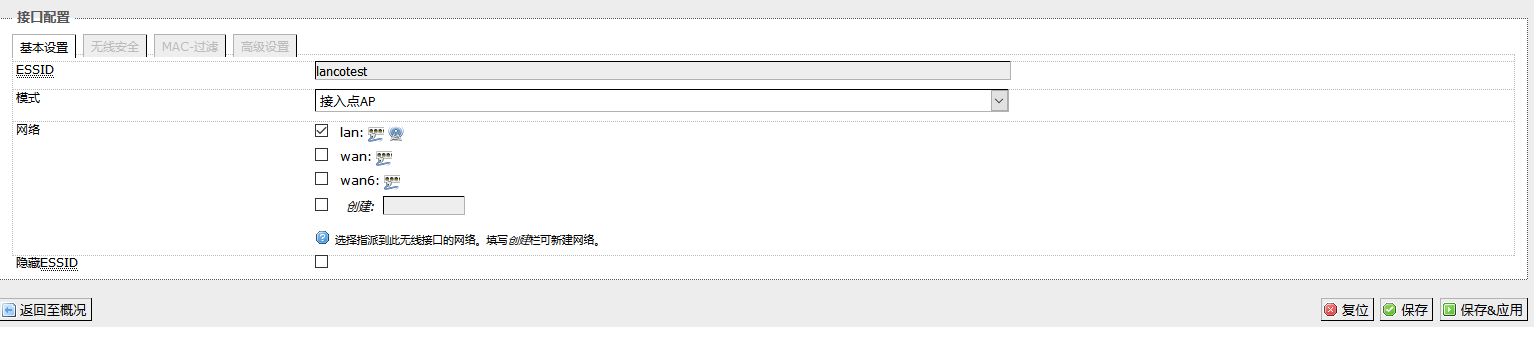
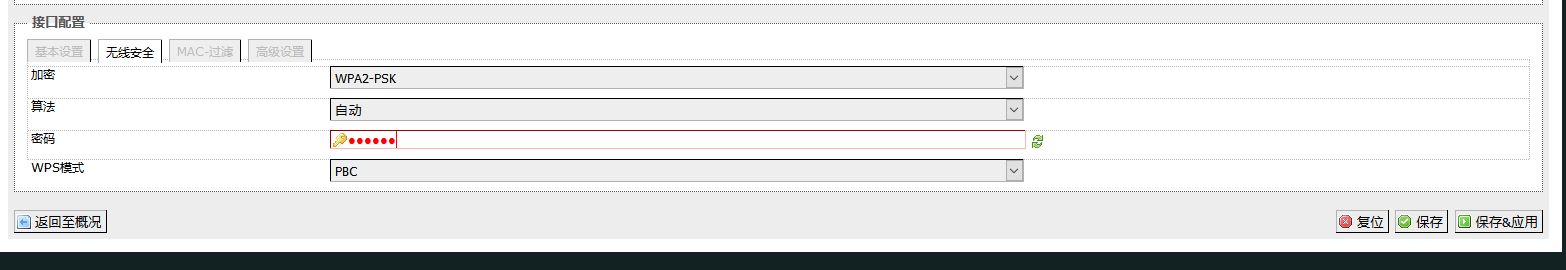


五，配置wifi，连接电源

A，配置wifi

连接网线后重启路由，使用浏览器登录192.168.1.1，进行wifi名称（essid）和密码（如图设置加密模式）设置





B，电源方案

需要24小时wifi的同学，建议使用一枚充电宝，为路由器供电同时连接充电宝充电接口进行充电，即在路由和电源间加一个充电宝，可以实现电源的稳定切换，不易破坏路由数据（某种闪断可能导致路由器配置丢失，从而需要重新设置，故应减少不稳定供电发生的频率）

附录：本破解方案涉及的所有开源代码及其协议

<https://github.com/drcoms/jlu-drcom-client>

作者：latyas

额外说明：在登录作者代码页的时候，发现更新了

|  |  |
| --- | --- |
| [python-mini\_client.py](https://github.com/drcoms/jlu-drcom-client/blob/master/python-mini_client.py) | [精简依赖的模块，安装python-mini即可](https://github.com/drcoms/jlu-drcom-client/commit/cf0033364e5c37086f85f37f9d41744ea5ff8a61) |

（https://github.com/drcoms/jlu-drcom-client/blob/master/python-mini\_client.py）

此版本由于python空间的精简，对5661a型号用户很有帮助，可以使用最新版本的pandorabox固件提升稳定性（附在下载包中），而之前由于空间占用问题，只能使用旧版本的pandorabox，新版目前没有发现过数据消失问题，具体方法同上，由于吉大于9月中旬停用旧版本drcom，可能伴随代码更新，故本贴暂保留原稳定方案，届时请关注进一步更新通知。

本教程附带client.py

import socket, struct, time

from hashlib import md5

import sys

import urllib2

class ChallengeException (Exception):

def \_\_init\_\_(self):

pass

class LoginException (Exception):

def \_\_init\_\_(self):

pass

s = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)

s.bind(("0.0.0.0", 61440))

s.settimeout(3)

SALT = ''

UNLIMITED\_RETRY = True

EXCEPTION = False

DEBUG = True

server = "10.100.61.3" # "auth.jlu.edu.cn"

username = "xxxxx"

password = "xxxxxx"

host\_name = "LIYUANYUAN"

host\_os = "8089D"

mac = 0x102AB37EA532

def challenge(svr,ran):

while True:

t = struct.pack("<H", int(ran)%(0xFFFF))

s.sendto("\x01\x02"+t+"\x09"+"\x00"\*15, (svr, 61440))

try:

data, address = s.recvfrom(1024)

except:

if DEBUG:

print '[challenge] timeout, retrying...'

continue

if address == (svr, 61440):

break;

else:

continue

if DEBUG:

print '[DEBUG] challenge:\n' + data.encode('hex')

if data[0] != '\x02':

raise ChallengeException

print '[challenge] challenge packet sent.'

return data[4:8]

def md5sum(s):

m = md5()

m.update(s)

return m.digest()

def dump(n):

s = '%x' % n

if len(s) & 1:

s = '0' + s

return s.decode('hex')

def ror(md5, pwd):

ret = ''

for i in range(len(pwd)):

x = ord(md5[i]) ^ ord(pwd[i])

ret += chr(((x<<3)&0xFF) + (x>>5))

return ret

def keep\_alive\_package\_builder(number,random,tail,type=1,first=False):

data = '\x07'+ chr(number) + '\x28\x00\x0b' + chr(type)

if first :

data += '\x0f\x27'

else:

data += '\xdc\02'

data += random + '\x00' \* 6

data += tail

data += '\x00' \* 4

#data += struct.pack("!H",0xdc02)

if type == 3:

foo = '\x31\x8c\x21\x3e' #CONSTANT

#CRC

crc = packet\_CRC(data+foo)

data += struct.pack("!I",crc) + foo + '\x00' \* 8

else: #packet type = 1

data += '\x00' \* 16

return data

def packet\_CRC(s):

ret = 0

for i in re.findall('..', s):

ret ^= struct.unpack('>h', i)[0]

ret &= 0xFFFF

ret = ret \* 0x2c7

return ret

def keep\_alive2():

#first keep\_alive:

#number = number (mod 7)

#status = 1: first packet user sended

# 2: first packet user recieved

# 3: 2nd packet user sended

# 4: 2nd packet user recieved

# Codes for test

tail = ''

packet = ''

svr = server

import random

ran = random.randint(0,0xFFFF)

ran += random.randint(1,10)

packet = keep\_alive\_package\_builder(0,dump(ran),'\x00'\*4,1,True)

s.sendto(packet, (svr, 61440))

data, address = s.recvfrom(1024)

ran += random.randint(1,10)

packet = keep\_alive\_package\_builder(1,dump(ran),'\x00'\*4,1,False)

s.sendto(packet, (svr, 61440))

data, address = s.recvfrom(1024)

tail = data[16:20]

ran += random.randint(1,10)

packet = keep\_alive\_package\_builder(2,dump(ran),tail,3,False)

s.sendto(packet, (svr, 61440))

data, address = s.recvfrom(1024)

tail = data[16:20]

print "[keep-alive2] keep-alive2 loop was in daemon."

i = 1

while True:

try:

time.sleep(5)

ran += random.randint(1,10)

packet = keep\_alive\_package\_builder(2,dump(ran),tail,1,False)

#print 'DEBUG: keep\_alive2,packet 4\n',packet.encode('hex')

s.sendto(packet, (svr, 61440))

data, address = s.recvfrom(1024)

tail = data[16:20]

#print 'DEBUG: keep\_alive2,packet 4 return\n',data.encode('hex')

ran += random.randint(1,10)

packet = keep\_alive\_package\_builder(2,dump(ran),tail,3,False)

#print 'DEBUG: keep\_alive2,packet 5\n',packet.encode('hex')

s.sendto(packet, (svr, 61440))

data, address = s.recvfrom(1024)

tail = data[16:20]

#print 'DEBUG: keep\_alive2,packet 5 return\n',data.encode('hex')

i = i+1

i = i % (0xFF)

check\_online = urllib2.urlopen('http://10.100.61.3')

foo = check\_online.read()

if 'login.jlu.edu.cn' in foo:

print '[keep\_alive2] offline.relogin...'

break;

#MODIFIED END

'''

if i % 10 == 0:

check\_online = urllib2.urlopen('http://10.100.61.3')

foo = check\_online.read()

if 'login.jlu.edu.cn' in foo:

print '[keep\_alive2] offline.relogin...'

break;

'''

except:

pass

import re

def checksum(s):

ret = 1234

for i in re.findall('....', s):

ret ^= int(i[::-1].encode('hex'), 16)

ret = (1968 \* ret) & 0xffffffff

return struct.pack('<I', ret)

def mkpkt(salt, usr, pwd, mac):

data = '\x03\01\x00'+chr(len(usr)+20)

data += md5sum('\x03\x01'+salt+pwd)

data += usr.ljust(36, '\x00')

data += '\x00\x00'

data += dump(int(data[4:10].encode('hex'),16)^mac).rjust(6,'\x00')

data += md5sum("\x01" + pwd + salt + '\x00'\*4)

data += '\x01\x31\x8c\x31\x4e' + '\00'\*12

data += md5sum(data + '\x14\x00\x07\x0b')[:8] + '\x01'+'\x00'\*4

data += host\_name.ljust(71, '\x00')

data += '\x01' + host\_os.ljust(128, '\x00')

data += '\x6d\x00\x00'+chr(len(pwd))

data += ror(md5sum('\x03\x01'+salt+pwd), pwd)

data += '\x02\x0c'

data += checksum(data+'\x01\x26\x07\x11\x00\x00'+dump(mac))

data += "\x00\x00" + dump(mac)

return data

def login(usr, pwd, svr):

import random

global SALT

i = 0

while True:

try:

try:

salt = challenge(svr,time.time()+random.randint(0xF,0xFF))

except ChallengeException:

if DEBUG:

print 'challenge packet exception'

continue

SALT = salt

packet = mkpkt(salt, usr, pwd, mac)

s.sendto(packet, (svr, 61440))

data, address = s.recvfrom(1024)

except:

print "[login] recvfrom timeout,retrying..."

continue

print '[login] packet sent.'

if address == (svr, 61440):

if data[0] == '\x05' and i >= 5 and UNLIMITED\_RETRY == False:

print '[login] wrong password, retried ' + str(i) +' times.'

sys.exit(1)

elif data[0] == '\x05':

print "[login] wrong password."

i = i + 1

time.sleep(i\*1.618)

elif data[0] != '\x04':

print "[login] server return exception.retry"

if DEBUG:

print '[login] last packet server returned:\n' + data.encode('hex')

time.sleep(1)

raise LoginException

continue;

break;

else:

if i >= 5 and UNLIMITED\_RETRY == False :

print '[login] packet received error, maybe you are under attacking'

sys.exit(1)

else:

i = i + 1

print '[login] package error, retrying...'

print '[login] login sent'

return data[-22:-6]

import httplib

def info(ip):

c = httplib.HTTPConnection(ip, 80, timeout=10)

c.request("GET", "")

r = c.getresponse()

if r.status != 200:

return None

s = r.read()

data = dict()

data["flux"] = int(s[s.index("flow='")+6:s.index("';fsele=")])

data["time"] = int(s[s.index("time='")+6:s.index("';flow")])

return data

def keep\_alive1(salt,tail,pwd,svr):

foo = struct.pack('!H',int(time.time())%0xFFFF)

data = '\xff' + md5sum('\x03\x01'+salt+pwd) + '\x00\x00\x00'

data += tail

data += foo + '\x00\x00\x00\x00'

print '[keep\_alive1] keep\_alive1,sent'

s.sendto(data, (svr, 61440))

try:

data, address = s.recvfrom(1024)

print '[keep\_alive1] keep\_alive1,server received'

except:

print '[keep\_alive1] Timeout!'

pass

def main():

print "auth svr:"+server+"\nusername:"+username+"\npassword:"+"\*\*\*\*\*\*\*\*"+"\nmac:"+str(hex(mac))

print "os:MSDOS 8.0"+"\nhostname: localhost"

print "DrCOM Auth Router Ver 1.2"

print "Version feature:\n[1] Auto Anti droping connection\n[2] Stronger exception handling."

while True:

try:

package\_tail = login(username, password, server)

except LoginException:

continue

keep\_alive2()

if \_\_name\_\_ == "\_\_main\_\_":

main()