The MITM Wifi Router with Raspberry Pi 3 b+

Conor Flynn

Outreach Project

1 November 2018
Introduction

Wifi is ubiquitous. A commodity and amenity so highly praised and utilized in current society that without it, one feels a deprivation of connection to the world. Most however, overlook the vast complexities of how Wifi works, the intricate security measures in place to prevent people with malicious intents from stealing your data, and the breaches in security (both physical and virtual) which are constantly under strain from attackers. Yet regardless of the amounts of security, people still are always the weakest link in a security system, which is known as the field of social engineering. One of the most forefront areas in cybersecurity is social engineering, in which the hackers mentally manipulate their victims to perpetrate what they desire. Here, the most a security system can do is warn the user of their actions, yet that does not fully insure the safety of the device/network. Social engineering is crucial to a “man in the middle attack” (MITM).

A MITM is the falsification of a trusted wifi router, in which the perpetrator uses to steal passing traffic. Yet the attacker cannot connect the user to the network, which is where social engineering is prevalent. The attacker will mimic the SSID (service set identifier) of the trusted network, enticing
the user to connect to theirs. Systems such as Windows warn of this, yet it is often subtle and obscure to the typical user. Windows will display a simple shield with an exclamation mark warning that the network does not meet all of the security protocols typically used. Users are often oblivious to security warnings like these, allowing for attackers to take advantage of them. This guide on how to build and construct a Raspberry Pi MITM router serves as an instruction manual as well as an awareness raiser for attacks such as this.

1 https://answers.microsoft.com/en-us/windows/forum/windows_10-networking/wifi-network-icon-with-exclamation-point/143888a4-4cac-4a4c-9fde-91d1a0c0dc66?auth=1
Materials:

- Raspberry Pi 3 b+ (preferably as a kit including: power-cable, HDMI-cable, preloaded SD card)
- Monitor
- Keyboard
- Mouse
- SD Card 4+ GB
  - Required if it did not come with the kit. Preferably with Raspbian OS installed. If it is not installed follow this guide²

Skills Required:

- Basic directory navigation skills
- Ability to type
- Basic understanding of variables and how to set them (i.e. \( a = 1 \))
- Ability to set up a barebone computer system (i.e. monitor, keyboard, mouse, power, and ethernet)
- Understanding of incoming and outgoing network traffic
  (only required for MITM portion)
Instructions (For Wifi Router Configuration):

Source: Medium

Notes:
- This methodology is NOT mine, I am merely explaining it in my own manner
- Additions ARE mine, including research and errors I came across while writing this
- Code supposed to be typed will be in the format of <sudo apt-get update> with the line of code actually being sudo apt-get update (When writing this line DO NOT write the < and > since they are there to help direct the reader).
- Text inside of a file will be denoted as [SAMPLE TEXT]
- If you want a short explanation

1. Complete the installation of Raspbian.
   a. NOOBS will guide the user through a short series of instructions upon the first boot-up of the pi
   b. Once there are no more instructions on the desktop you know you are complete.

2. Update check / DHCP server installation
   a. <sudo apt-get update>
      i. Makes sure that all software on the pi is up to date

b. `<sudo apt-get install hostapd isc-dhcp-server>`
   
i. Installs the programs going to be used (hostapd and DHCP server)

3. Make sure that a eth0 and wlan0 exist on the Pi
   
a. `<ifconfig -a>`
   
b. It will state it on the left side along with other things such as lo

4. DHCP server configuration
   
a. `<sudo cp /etc/dhcp/dhcpd.conf /etc/dhcp/dhcpd.conf.default>`
   
i. Writes a backup of the dhcpd configuration file
   
b. `<sudo nano /etc/dhcp/dhcpd.conf>`
   
i. Opens text editor in “nano”
   
ii. Comment out the following lines
   
1. `[option domain-name “example.org”;]`
   
2. `[option domain-name-servers ns1.example.org, ns2.example.org;]`
   
a. To comment out a line place a [#] in front of the line
   
b. i.e. [#option domain-name “example.org”;]
   
iii. Uncomment the line that says [#authoritative;]
1. It is under the lines that say

2. [#If this DHCP server is the official DHCP server for the local network, the authoritative directive should be uncommented.]

4. Should now read [authoritative;]

iv. Add this to the bottom of the file (recommended to copy/paste this)

v. [ subnet 192.168.42.0 netmask 255.255.255.0 {
range 192.168.42.10 192.168.42.50;
option broadcast-address 192.168.42.255;
option routers 192.168.42.1;
default-lease-time 600;
max-lease-time 7200;
option domain-name "local";
option domain-name-servers 8.8.8.8, 8.8.4.4;
} ]

1. Remember DO NOT include the [ and ] at the beginning and end

2. Line explanations
   a. [range 192.168.42.10 192.168.42.50]
i. Assigns the subnet
   192.168.42.10-50

ii. Allows for 40 devices

b. [option broadcast-address
   192.168.42.255]

i. Sets broadcast address for
   connected devices

c. [option domain-name-servers]

i. Assigns Google’s public DNS
   (8.8.8.8 and 8.8.4.4)

3. NOTE: to save a file use the following key commands

   a. CTRL+O

      i. Writes out to the file

   b. Enter

      i. Saves the new document

   c. CTRL+X

      i. Exits the file

vi. To paste in the terminal right click and select paste

1. Note that sometimes using CTRL+V will result in unwanted text
c. `<sudo nano /etc/default/isc-dhcp-server>`
   
   i. Edit the line `[INTERFACES=""]` to read
      
      `[INTERFACES="wlan0"]`
   
   ii. NOTE:
      
      1. Sometimes the lines look like
         
         a. `[INTERFACESIPV4=""
            INTERFACEIPV6=""]`
      
      2. Change to
         
         a. `[INTERFACESIPV4="wlan0"
            INTERFACEIPV6="wlan0"
            INTERFACES="wlan0"]`

5. Set up wlan0

   a. `<sudo ifdown wlan0>`
      
      i. Makes sure wlan0 is down

   b. `<sudo cp /etc/network/interfaces
       /etc/network/interfaces.backup>`
      
      i. Creates a backup file

   c. `<sudo nano /etc/network/interfaces>`
      
      i. Opens the network interface file

      ii. Edit the file to read exactly as follows:

      1. `[auto lo`
iface lo inet loopback
iface eth0 inet dhcp
allow-hotplug wlan0

iface wlan0 inet static
address 192.168.42.1
netmask 255.255.255.0
post-up iw dev $IFACE set power_save off

2. Close and save the file
   a. CTRL+O
   b. Enter
   c. CTRL+X
   d. <sudo ifconfig wlan0 192.168.42.1>
      i. Assigns the router a static ip

6. Hostapd Configuration
   a. <sudo nano /etc/hostapd/hostapd.conf>
      i. Opens the hostapd configuration file
   ii. IMPORTANT:
      1. To edit the name of your network change the
         parameter [ssid=WiPi]

https://www.techopedia.com/definition/9544/static-internet-protocol-ip-address-static-ip-address
a. Default name is WiPi  
b. I.e. [ssid=TEST]

2. To edit the password of your network change the parameter [wpa_passphrase=xyz]

   a. Default password is xyz  
   b. I.e. [wpa_passphrase=PASSWORD]

b. If needed to set a driver ID (not used on many systems)

   i. Do not do this step if it does not apply  
   ii. <basename $( readlink /sys/class/net/wla0/device/driver )>

7. NAT (Network address translation)

   a. <sudo cp /etc/sysctl.conf /etc/sysctl.conf.backup>

      i. Creates a backup file

   b. <sudo nano /etc/sysctl.conf>

      i. Opens config file in nano  
      ii. Add the following line to the bottom of the file

         1. [net.ipv4.ip_forward=1]

      iii. Save and close the file

   c. <sudo sh -c “echo 1 > /proc/sys/net/ipv4/ip_forward”>

      i. Activates the modified code immediately
d. Modify iptables to create a network translation between eth0 and wlan0
   
   i. <sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE>
   
   ii. <sudo iptables -A FORWARD -i eth0 -o wlan0 -m state --state RELATED,ESTABLISHED -j ACCEPT>
   
   iii. <sudo iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT>

   e. <sudo sh -c “iptables-save > /etc/iptables.ipv4.nat”>
      
   i. Runs program on startup

   f. **<sudo nano /etc/network/interfaces up iptables-restore < /etc/iptables.ipv4.nat>

8. Startup
   
   a. <sudo /usr/sbin/hostapd /etc/hostapd/hostapd.conf>
      
   i. Program can always be stopped with CTRL+d

   b. Starting network
      
   i. <sudo service hostapd start>
   
   ii. <sudo service isc-dhcp-server start>

   c. Testing if it is running
      
   i. <sudo service hostapd status>
   
   ii. <sudo service isc-dhcp-server status>

   d. Allowing for program to run at boot time
i.  <sudo update-rc.d hostapd enable>

ii.  <sudo update-rc.d isc-dhcp-server enable>

e. Rebooting for system check

   i.  <sudo reboot>

9. Data Monitoring

   a.  <tail -f /var/log/syslog>
Errors Encountered

1. Installing a clean version of NOOBS (NOTE: this is used in most of the solutions)
   a. Download NOOBS image
      i. Do NOT download the Lite version
      ii. Download the zip file format
   b. Clean the SD card (remove all files)
   c. Extract the image onto the SD card
   d. Plug it back into Raspberry Pi
   e. Note this erases ALL progress on your project meaning you have to start from the beginning again

2. Wifi connection lost and would not reconnect
   a. Solution - Reinstall Raspbian

3. Raspberry Pi will not recognize SD card
   a. Solution 1 (Reformat card)
      i. For Windows
         1. Open file explorer
         2. Right click on the SD Card (Should be either the F:\ or E:\ drives)
         3. Click format

4. Make sure that the format is set to default (FAT32 or FAT)
5. Do not select quick format
6. NOTE: This will remove ALL progress from your project
7. Reinstall Raspbian

ii. For other operating systems (works on Windows as well)
   1. Download SD Memory Card Formatter\(^6\)
   2. Follow the setup instructions
   3. Run and select FAT32 or FAT
   4. NOTE: This will remove ALL progress from your project
   5. Reinstall Raspbian

b. Solution 2 (Create a bootable USB drive)
   i. For Windows
      1. Plug in flash drive to computer (NOT Raspberry Pi)
      2. Open Command Prompt\(^7\)
      3. Run the command `<diskpart>
         a. Opens a new Windows

---
\(^6\) [https://www.sdcard.org/downloads/formatter_4/](https://www.sdcard.org/downloads/formatter_4/)
\(^7\) [https://www.digitalcitizen.life/7-ways-launch-command-prompt-windows-7-windows-8](https://www.digitalcitizen.life/7-ways-launch-command-prompt-windows-7-windows-8)
4. `<list disk>`
   a. Take note of which disk is the flash drive (Most likely disk 1)

5. `<select disk DISK_NUMBER>`
   a. i.e. `<select disk 1>`

6. `<clean>`
   a. NOTE: This will remove ALL progress from your project

7. `<create part pri>`

8. `<select part 1>`

9. `<format fs=fat32 quick>`

10. `<active>`

11. `<exit>`

12. Reinstall Raspbian onto the flash drive

13. Plug in the flash drive into the Raspberry Pi
   a. Make sure the SD Card is NOT plugged into the Raspberry Pi
   b. The NOOBS OS installation screen should appear after a couple minutes.
Works Cited


“SD Memory Card Formatter 5.0 for SD/SDHC/SDXC.” SD Association, www.sdcard.org/downloads/formatter_4/.
