Raspberry Pi Secure Router

Part 1: Firewall

What are we doing?

- Creating a secure router using a Raspberry Pi
- What do you mean?
  - When using a public network, I want to ensure all my traffic uses a VPN.
    - That means my traffic comes out to the Internet from a distant location
  - I realize I can run a VPN on my laptop
    - I do… I current run proXPN
  - But, not all my devices support VPN connections
    - e.g., Apple TV
- So, let's build a device that will provide a secure tunnel

Topology

- I'd like the Raspberry Pi Secure Router to support a variety of configurations
  - My devices attaches to the Ethernet port on the Raspberry Pi then the Raspberry Pi connects to an untrusted wireless network
    - This is the most secure
  - My device attaches to the Raspberry Pi using WiFi then the Raspberry Pi connects to an untrusted wired network.
    - The traffic between my device and the Raspberry Pi could be snooped
  - My device attaches to the Raspberry Pi using one WiFi dongle then the Raspberry Pi connects to an untrusted wireless network using another WiFi dongle.
    - The traffic between my device and the Raspberry Pi could be snooped
- We'll develop only the first configuration in this class, but the others naturally follow

What are the big tasks?

- Part 1 – Configure the Raspberry Pi as a firewall
- Part 2 – Configure a VPN tunnel
- Part 3 – Configure a web server and a way to configure the Raspberry Pi's wireless networks in headless mode

Part 1

- Configure the Raspberry Pi as a firewall
  - Initialize the Raspberry Pi
  - Configure the Raspberry Pi WiFi dongle to attach to a trusted wireless network
  - Configure the Raspberry Pi as a DHCP server
    - Configure a fixed IP address for the Raspberry Pi
    - Install and configure a DHCP server
  - Configure the Raspberry Pi routing tables
  - Configure Raspberry Pi WiFi dongle to attach to the untrusted wireless network

Part 1.1

- Initialize the Raspberry Pi
  - Load the Raspberry Pi with Raspian Jessie, March 2016
  - Expand the file system
  - Configure locale, timezone, keyboard, WiFi country
  - Configure boot to console, with login
  - Change the pi password
  - Connect to a trusted wireless network
  - “sudo apt-get update”
  - “sudo apt-get upgrade”

Part 1.2

- Configure the raspberry PI as a DHCP server
  - Modify /etc/network/interfaces. Replace:

    iface etho inet local

    -with-

    auto etho
    iface etho inet static
    address 192.168.14.1
    netmask 255.255.255.0
  - Reboot
  - "sudo apt-get install isc-dhcp-server"
  - Modify /etc/dhcp/dhcpd.conf. Comment out:

    option domain-name "example.org"
    option domain-name-servers ns1.example1.org, ns2.example.org
  - Modify /etc/dhcp/dhcpd.conf. Uncomment:

    #authoritative

Part 1.2 (cont)

- Configure the raspberry PI as a DHCP server (cont)
  - Modify /etc/dhcp/dhcpd.conf. Append:

```
subnet 192.168.14.0 netmask 255.255.255.0 {
  range 192.168.14.10 192.168.14.50;
  option broadcast-address 192.168.14.255;
  option routers 192.168.14.1;
  default-lease-time 600;
  max-lease-time 7200;
  option domain-name "local";
  option domain-name-servers 8.8.8.8, 8.8.4.4;
}
```

  - Modify /etc/default/isc-dhcp-server Replace:

    INTERFACES=""

    -with-

    INTERFACES="etho"

  - "sudo service isc-dhcp-server start"
  - "sudo service isc-dhcp-server status"
  - "sudo update-rc.d isc-dhcp-server enable"
  - Reboot and test

Part 1.3

- Configure the Raspberry Pi network routing
  - Modify /etc/sysctl.conf. Uncomment:

    #net.ipv4.ip_forward=1
  - Reboot
  - "sudo iptables -t nat -A POSTROUTING -o wlano -j MASQUERADE"
  - "sudo iptables -A FORWARD -i wlano -o etho -m state --state RELATED,ESTABLISHED -j ACCEPT"
  - "sudo iptables -A FORWARD -i etho -o wlano -j ACCEPT"
  - "sudo iptables-save > /tmp/t1"
  - "sudo mv /tmp/t1 /etc/iptables.ipv4.nat"
  - Modify /etc/network/interfaces. Append:

    up iptables-restore < /etc/iptables.ipv4.nat
  - Reboot
  - Test Routing
- Connect to the untrusted wireless network

What's Next?

- Part 2 – Configure the VPN tunnel