Install and configure a Debian based UniFi controller

1. Configuring Debian

First you will need to download the correct Debian image for your architecture. There are generally two images used, a smaller installation image or a larger complete installation image. The smaller image is a bare bones Debian that requires an internet connection during the installation. The larger installation image contains most of the popular packages and does not require an internet connection during installation. Since this installation will be used only as a UniFi controller we will be going with the small installation image. Both images can be downloaded here:

http://www.debian.org/distrib

We will not be covering the installation of Debian in this guide since it is well documented elsewhere.

http://www.debian.org/releases/stable/installmanual

Once Debian is up and running, the first thing we will do is assign it a static IP address. We do this by editing the /etc/network/interfaces file using the Nano editor. Type the following command to open up the interfaces file in Nano:

```
# nano /etc/network/interfaces
```

Now we will edit the eth0 interface (or whichever interface you are using) with the static IP information (obviously substitute this info with your IP and subnet information):

```
auto eth0
iface eth0 inet static
        address 192.168.1.15
        netmask 255.255.255.0
        gateway 192.168.1.1
        network 192.168.1.0
        broadcast 192.168.1.255
        dns-nameservers 192.168.1.1 8.8.8.8
```

Next we will install SSH for remote access to the controller. Prior to installing the OpenSSH server we will want to update the packages database. You do this by running the following command as the root user:

```
# apt-get update
```

Now we can install the OpenSSH server with the following command:

```
# apt-get install openssh-server
```

That's it. SSH should now be running and accepting connections. You can test this by SSH'ing locally with the following command:

```
# ssh root@localhost
```

2. Installing the UniFi controller software

Now on to installing the UniFi software. First we will need to add the Ubiquiti source to our sources list. This will tell Debian to look for packages available from Ubiquiti for installation. To edit the sources list, open the sources.list file in Nano with the following command (as root user):

```
# nano /etc/apt/sources.list
```

Now add the following lines below the existing sources and save the changes:

```
# Ubiquiti UniFi updates
deb http://www.ubnt.com/downloads/unifi/distros/deb/debian debian ubiquiti
```

Next we will need to add the Ubiquiti GPG keys. Type the following commands as root user:

```
# apt-key adv --keyserver keyserver.ubuntu.com --recv C0A52C50
```

and

```
# apt-key adv --keyserver keyserver.ubuntu.com --recv 7F0CEB10
```

And finally we need to install the UniFi controller software. First we need to update our package database again, which will now include the UniFi package information from Ubiquiti.

```
# apt-get update
```

And last but not least, installation of the actual UniFi package. You can choose to install the stable version or the latest beta version:

```
# apt-get install unifi
```

or

```
# apt-get install unifi-beta
```

The UniFi controller software should now be accessible by visiting the following URL (replacing the IP address with whatever you used):

```
https://192.168.1.15:8443
```

3. Configuring the firewall

The final step is to configure the controller's firewall to only allow SSH and the UniFi ports. This step is optional, but recommended, especially if you chose to do a full install of Debian since it will have a much larger attack surface. First we will create the file to store the firewall rules in. Run the following command as root user:

```
# nano /etc/network/firewall-rules
```

Now paste the following set of rules into the file and save it (make sure and change eth0 to whatever interface you are using):

```
*filter

# Flush any existing rules
-F

# Default policy: drop all inbound and allow all outbound
-P FORWARD DROP
-P INPUT   DROP
-P OUTPUT  ACCEPT

# Accepts all established inbound connections
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# Allow local programs that use loopback (Unix sockets)
-A INPUT -s 127.0.0.0/8 -d 127.0.0.0/8 -i lo -j ACCEPT

# Allow ICMP pings
-A INPUT -i eth0 -p icmp --icmp-type echo-request -j ACCEPT

# Allow SSH from anywhere
-A INPUT -i eth0 -p tcp --dport 22 -j ACCEPT

# Allow all the UniFi ports from anywhere
-A INPUT -i eth0 -p tcp --dport 8080 -j ACCEPT
-A INPUT -i eth0 -p tcp --dport 8081 -j ACCEPT
-A INPUT -i eth0 -p tcp --dport 8443 -j ACCEPT
-A INPUT -i eth0 -p tcp --dport 8880 -j ACCEPT
-A INPUT -i eth0 -p tcp --dport 8843 -j ACCEPT
-A INPUT -i eth0 -p udp --dport 3478 -j ACCEPT
-A INPUT -i eth0 -p udp --dport 10001 -j ACCEPT

COMMIT
```

Next we need to set Debian to run these rules when the machine starts up. We will do that by calling firewall-rules from the /etc/network/interfaces file. We will need to add the following line to the bottom of the interface's config:

```
post-up iptables-restore < /etc/network/firewall-rules
```

It should look like this:

```
auto eth0
iface eth0 inet static
        address 172.24.32.20
        netmask 255.255.255.0
        gateway 172.24.32.1
        network 172.24.32.0
        broadcast 172.24.32.255
        dns-nameservers 172.24.32.1 8.8.8.8
        post-up iptables-restore < /etc/network/firewall-rules
```

Now you can either reboot the machine or restart the interface to activate the firewall rules. You can restart the interface with the following command:

```
# ifdown eth0 && ifup eth0
```

Verify that the firewall rules are loaded with the following command:

```
# iptables -L
```

It should look like this:

```
Chain INPUT (policy DROP)
target     prot  opt      source           destination
ACCEPT     all     --      anywhere         anywhere         state RELATED,ESTABLISHED
ACCEPT     all     --      loopback/8       loopback/8
ACCEPT     icmp  --       anywhere         anywhere         icmp echo-request
ACCEPT     tcp     --      anywhere         anywhere         tcp dpt:ssh
ACCEPT     tcp     --      anywhere         anywhere         tcp dpt:http-alt
ACCEPT     tcp     --      anywhere         anywhere         tcp dpt:tproxy
ACCEPT     tcp     --      anywhere         anywhere         tcp dpt:8443
ACCEPT     tcp     --      anywhere         anywhere         tcp dpt:8880
ACCEPT     tcp    --       anywhere         anywhere         tcp dpt:8843
ACCEPT     udp     --      anywhere         anywhere         udp dpt:3478
ACCEPT     udp     --      anywhere         anywhere         udp dpt:10001

Chain FORWARD (policy DROP)
target    prot    opt      source          destination

Chain OUTPUT (policy ACCEPT)
target    prot    opt      source          destination
```

And that's it! You've got yourself a solid, secure, small footprint UniFi controller.