How to Image and Program the [Product Name]

Install nmap on Mac (use .dmg installer).

A PoE switch has been setup at the warehouse to facilitate imaging [Product]. Physically connect a [Product] to the PoE switch and run the following command using nmap:

```
sudo nmap -sP -n 172.16.00.0/24
```

Warehouse HP Switch IP Address: ***172.16.00.0/24***

Pipe using grep to get a more useful command response:

```
sudo nmap -sP -n 172.16.0.0/24 | grep -X 2 Raspberry
```

Note: the -B option and the -A option can be used to get the lines before and after the target line.

You should see a response like this:

```
$ sudo nmap -sP -n 172.16.00.0/24 | grep Raspberry
MAC Address: B8:27:EB:EC:C1:4F (Raspberry Pi Foundation)
MAC Address: B8:27:EB:77:0A:C8 (Raspberry Pi Foundation)
MAC Address: B8:27:EB:5B:F2:1E (Raspberry Pi Foundation)
MAC Address: B8:27:EB:A1:73:9E (Raspberry Pi Foundation)
MAC Address: B8:27:EB:B1:9E:B4 (Raspberry Pi Foundation)
MAC Address: B8:27:EB:14:AF:9A (Raspberry Pi Foundation)
MAC Address: B8:27:EB:ED:57:B8 (Raspberry Pi Foundation)
MAC Address: B8:27:EB:83:A3:51 (Raspberry Pi Foundation)
MAC Address: B8:27:EB:FF:E2:92 (Raspberry Pi Foundation)
MAC Address: B8:27:EB:C4:14:95 (Raspberry Pi Foundation)
```

Or this with the -B option used to get the two preceding lines (which includes the [Product]'s IP address):

```
Nmap scan report for 172.16.00.31
Host is up (0.00047s latency).
MAC Address: B8:27:EB:ED:57:B8 (Raspberry Pi Foundation)
--
Nmap scan report for 172.16.00.32
Host is up (0.00011s latency).
MAC Address: B8:27:EB:83:A3:51 (Raspberry Pi Foundation)
--
Nmap scan report for 172.16.00.33
Host is up (0.00011s latency).
MAC Address: B8:27:EB:FF:E2:92 (Raspberry Pi Foundation)
```

IMPORTANT: the warehouse WiFi and ethernet are on different segments, so your Mac must be connected to ethernet to see the MAC addresses.

Alternative WiFi Method (if you don’t have ethernet access)

An alternative method is to run the following nmap command:

```
sudo nmap -p 22 –script=banner 172.16.00.0/24
```

Look for SSH banners with “raspbian” in them. SSH into each one until you find the MAC you’re looking for. Example response:

```
Nmap scan report for 172.16.00.24
Host is up (0.022s latency).

PORT STATE SERVICE
22/tcp open ssh
|_banner: SSH-2.0-OpenSSH_9.7p1 Raspbian-5+debwu3
```

Raspberry Pi: Factory Default Username and Password

Username: pi
Password: raspberry

The IP address for this NT0000 is given (172.16.00.24). Use this IP address to SSH into the NT0000:

```
ssh pi@172.16.0.24
```

IMPORTANT: after running Salt Sync, the user “pi” is deleted, so you must login using your own account:

```
ssh rfrei@172.16.0.24
```

You will be presented with the {Product] factory test screen:

Select “Exit” to enter the shell.

You can check the MAC address of the [Product] using ifconfig:

The MAC address for this [Product]is shown under eth0: **b8:27:eb:09:c2:bd**

Once you’ve confirmed the MAC address, you can use secure copy to the [Product]:

```
scp ~/Desktop/bootstrap_[product].sh pi@172.16.00.24:/home/pi
```

bootstrap_gateway.sh

```
#!/usr/bin/env bash


PROVISION_API='http://provision.telkonet.com'
if [ ! -f /etc/provision_complete ]; then
curl -L $PROVISION_API/gateway/bootstrap > /tmp/provision.py


python3 /tmp/provision.py \
--provision-api $PROVISION_API
fi
```

Now run the shell script:

```
sudo bash bootstrap_gateway.sh
```

Running Salt Sync From the Salt Master

After connecting your NT0000s to the PoE Switch, go to Fleet Command > Salt States and click on the “Sync” button.

Note: the Sync button is shown running the Sync process in the image above. At the time of this writing, this process will spin forever in the web page. This is because the job takes longer than the HTTP timeout; 5-7 minutes should be sufficient for synchronizing ten gateways.

Automated Method for Testing the Web Interface on a Gateway

You can use this bash script to test the web interface very quickly (instead of doing the browser test):

```
for host in $(cat /Users/rfrei/Desktop/tmp/hosts); do
echo -e "\n$host"
curl http://$host:5000 > /dev/null 2>&1
if [ $? -eq 0 ]; then
ssh $host 'ip link show dev eth0 | grep link/ether'
fi
done
```

The corresponding “hosts” file must be populated with the corresponding NT0000 IPs:

```
172.16.00.21
172.16. 00.25
172.16.00.26
172.16. 00.27
172.16. 00.28
172.16. 00.29
172.16. 00.31
172.16. 00.32
172.16. 00.33
172.16. 00.36
```

This script will produce the following result:

```
$ bash nt0000_web_interface_test.sh
172.16.00.21
link/ether b8:27:eb:xl:c1:4f brd ff:ff:ff:ff:ff:ff


172.16.00.25


172.16..26
link/ether b8:27:eb:5b:f2:1e brd ff:ff:ff:ff:ff:ff


172.16.00.27
link/ether b8:27:eb:a1:73:9e brd ff:ff:ff:ff:ff:ff


172.16.00.28
link/ether b8:27:eb:b1:9e:b4 brd ff:ff:ff:ff:ff:ff


172.16.00.29
link/ether b8:27:eb:14:af:9a brd ff:ff:ff:ff:ff:ff


172.16.00.31
link/ether b8:27:eb:ed:57:b8 brd ff:ff:ff:ff:ff:ff


172.16.00.32
link/ether b8:27:eb:83:a3:51 brd ff:ff:ff:ff:ff:ff


172.16.00.33
link/ether b8:27:eb:ff:e2:92 brd ff:ff:ff:ff:ff:ff


172.16.00.36
link/ether b8:27:eb:c4:14:95 brd ff:ff:ff:ff:ff:ff
```

Note the failure in red. No link/ethernet response. These require re-imaging.

<end of sample document>