Python mDNS Utilities for Nix

Perry Kundert

June 5, 2019

Contents

1 Setting up your HoloPort OS

If you want to use the Holo testnet to perform this test across multiple nodes, you'll first need to ensure that `/var/lib/zerotier-one/` does contains an `identity.*` and `authtoken.secret` unique to this box (created when the VM was first run, not one generated at VM creation time, in this case `Jun 3 17:07`):

```
sudo ls -l /var/lib/zerotier-one/
total 44
-rw------- 1 root root    24 Jun  4 18:04 authtoken.secret
drwx------ 4 root root  4096 Jun  3 17:07 controller.d
-rw-r--r-- 1 root root   141 Jun  4 18:04 identity.public
-rw------- 1 root root   270 Jun  4 18:04 identity.secret
drwxr-xr-x 2 root root  4096 Jun  5 12:56 networks.d
drwxr-xr-x 2 root root 12288 Jun  5 04:13 peers.d
-rw-r--r-- 1 root root   634 Jun  3 17:07 planet
-rw-r--r-- 1 root root     5 Jun  4 18:04 zerotier-one.pid
-rw-r--r-- 1 root root     4 Jun  4 18:04 zerotier-one.port
```

If not, remove the `authtoken.secret` and `identity.*`, and reboot.

1.1 Registering your HoloPort OS

Register your HoloPort OS, using an email address, and get it activated by someone in Holo, so that it can join the testnet. Later in [Setting up ZeroTier], you should see `holohost_testnet` listed.

```
$ holo init
```

2 Networking Configuration

2.1 Opening Firewall Ports

mDNS uses UDP port 5353; open it by adding = networking.firewall.allowedUDPPorts = [5353];= to /etc/nixos/configuration.nix, and running `$ sudo nixos-rebuild switch`.

2.2 Setting up ZeroTier

To join a network:

```
# Join the earth.zerotier.net network, which probably has mDNS services registered
$ sudo zerotier-cli join 8056c2e21c000001
$ sudo zerotier-cli listnetworks
200 listnetworks <nwid> <name> <mac> <status> <type> <dev> <ZT assigned ips>
200 listnetworks 93afae5963c547f1 holohost_testnet f2:e5:c1:4d:22:1a OK PRIVATE ztzlggwhus fd93:afae:5963:c547:f199:93a2:042e:7b
200 listnetworks 8056c2e21c000001 earth.zerotier.net 02:a2:04:32:99:76 OK PUBLIC ztmjfmfyq5 fd80:56c2:e21c:0000:0199:93a2:042e:7
```

3 Using `python-zeroconf` on Nix

We're going to start a nix-shell configuration, with Python 3.5, virtualenv and pip available.

3.1 Getting Python onto Nix

Get the packages we'll need from github.com. Note that we're using a custom branch of python-zeroconf, to enable it to `--find` all available services.

```
$ mkdir -p ~/src && cd ~/src
$ git clone https://github.com/pjkundert/python-zeroconf -b browser-find
$ git clone https://github.com/pjkundert/mdns-python
```

3.2 Create a Python Virtual Environment

We want to make `python3` available whenever we need it, without being inside the nix-shell instance. We'll create a virtualenv containing links to the Python interpreter, and the various python modules needed by the python-zeroconf `registration.py` and `browser.py` utilities. First, run a nix-shell that installs and makes available the Python 3.5 interpreter, virtualenv and pip in nixos, and then create the virtualenv `venv` (you may need to re-run the nix-shell command later to refresh Python and its libraries, after updating the operating system with `sudo nixos-rebuild switch`):

```
$ cd ~/src/mdns-python
$ nix-shell

[nix-shell:~/src/mdns-python]$ virtualenv venv
[nix-shell:~/src/mdns-python]$ source venv/bin/activate
...
(venv)
[nix-shell:~/src/mdns-python]$ pip install ifaddr
...
(venv)
^D
$
```

3.3 Simulating an mDNS Service Registration

Later, whenever you log in and want to use Python w/ ifaddr, eg. to use python-zeroconf utilites like `registration.py`, to simulate an mDNS service advertisement:

```
$ source ~/src/mdns-python/venv/bin/activate
(venv)
$ PYTHONPATH=~/src/python-zeroconf/ ~/src/python-zeroconf/examples/registration.py &
```

3.4 Browsing for mDNS Services

To browse available networks for mDNS services; add `--find` to the `browser.py` command to search for all available services, or default to just `_http._tcp.local.`:

```
$ source ~/src/mdns-python/venv/bin/activate
(venv)
$ PYTHONPATH=~/src/python-zeroconf/ ~/src/python-zeroconf/examples/browser.py

Browsing 1 service(s), press Ctrl-C to exit...

Service LEDE-14._http._tcp.local. of type _http._tcp.local. state changed: ServiceStateChange.Added
Info from zeroconf.get_service_info: ServiceInfo(type='_http._tcp.local.', name='LEDE-14._http._tcp.local.', addresses=[b'\x1d\x
  Addresses: 29.20.140.13:80
  Weight: 0, priority: 0
  Server: LEDE-14.local.
  Properties are:
    b'path': b'/'
```