Team 76 (Brypt) Setup Guide

Part One: Desktop Application

Pick one of the following two testing methods:

1. Brypt Ubuntu Virtual Machine (Recommended)
    a. Difficulty: **Easy**
    b. Setup Time: Download Time + VM Install Time
    c. Requires: VMWare Workstation 15 or VMWare fusion and an external Ubuntu/Debian/Linux WiFi adapter (Confirmed working: TP-Link WN725n).
    d. Download the Brypt Ubuntu VM.
        i. https://drive.google.com/open?id=1qHn1LcWTNlIxZfp3BWeQqCVIhMY78lTz
        ii. https://oregonstate.box.com/s/vkcfl9doalik1khi3iwz2oixvxndgdr3
    e. Make a new Virtual Machine with the Brypt Ubuntu OVA using VMWare Workstation 15 or VMWare Fusion.
        i. The Brypt Ubuntu VM login credentials are username: “brypt-client” and password: “secured”.
    f. Ensure you can connect your external WiFi adapter directly to the VM.
    g. Disable the default shared/bridged internet connection to the VM.
    h. Connect to an Internet accessible WiFi Access Point.
    i. Follow Brypt Setup Guide Part Two before continuing.
    j. Open a terminal and run the development version of the Brypt Desktop.
        cd ~/brypt-desktop
        npm run dev
        i. Note: The application cache may think there’s two nodes connected. This is from the prior test run. The clear the nodes store, simply close the application and run again.
    k. Register for a new account or use the credentials: username: “piscitev” password “secured”.
    l. Follow Brypt Setup Guide Part Five to continue.

2. Manual Installation
    a. Difficulty: **Hard**
    b. Setup Time: 2-3 hours
    c. Install required libraries and resources.
        i. Requires: Ubuntu 18.04 LTS and an external Ubuntu/Debian/Linux WiFi adapter.
        ii. If running in a Virtual Machine follow steps *f* through *h* from Option 1 first.
        iii. NOTE: FontAwesome Pro files must be requested from Vincenzo Piscitello (piscitev@oregonstate.edu) and moved into the requisite folder. The desktop application WILL NOT work without these files.

d. Install the required libraries and resources:
   i. Basic Setup

      cd ~
      sudo apt update
      sudo apt upgrade
      sudo apt-get install libtool pkg-config build-essential autoconf automake git openssl libssl-dev libgonf-2-4 build-essential

   ii. Node.js/NPM
      1. https://nodejs.org/en/

      cd ~
      curl -sL https://deb.nodesource.com/setup_10.x | sudo -E bash -
      sudo apt-get install -y nodejs

   iii. ZeroMQ 4.x
      1. http://zeromq.org/intro:get-the-software
      2. Install the ZeroMQ C++ language bindings
         a. https://github.com/zeromq/cppzmq

      cd ~
      wget http://download.zeromq.org/zeromq-4.1.4.tar.gz
      tar -zxvf zeromq-4.1.4.tar.gz
      cd zeromq-4.1.4/
      ./configure --without-libsodium
      make
      sudo make install
      sudo ldconfig
      cd ~
      git clone https://github.com/zeromq/cppzmq.git
      sudo cp cppzmq/zmq.hpp /usr/local/include/

   iv. OpenSSL 1.1.0fg
      1. Note: Electron and Node.js use competing branches of OpenSSL; a static library must be installed onto your system.
         a. https://wiki.openssl.org/index.php/Compilation_and_Installation

            cd ~
            cd /tmp/
            wget http://www.openssl.org/source/openssl-1.1.0g.tar.gz
            tar -zxvf openssl-1.1.0g.tar.gz
            cd openssl-1.1.0g/
            make clean
            ./config --static -static -fPIC shared
            sudo make INSTALL_PREFIX=/tmp/package-root install

         b. You should now see "libcrypto.a" in the list of files using the command *ls /usr/local/lib/*

   v. Node-gyp

```
cd ~
npm install node-gyp -g
```

e. Clone and build the Brypt Desktop Git repository from Github.com.

```
cd ~
git clone https://github.com/vpiscitello/brypt-desktop.git
cd ~/brypt-desktop
git checkout ubuntu-64
npm install
node-gyp configure && node-gyp build
./node_modules/.bin/electron-rebuild
```

f. Follow Brypt Setup Guide Part Two before continuing.

g. Run the development version of the Electron application.

```
npm run dev
```

3. Start your Brypt network and login to the desktop application.
    a. Register for a new account or use the credentials: username: “piscitev” password “secured”
4. **Follow Brypt Setup Guide Part Five to continue.**

Part Two: Brypt Network - General Purpose Nodes

You have two options: flash the Raspberry Pi and run our installation script (more difficult), or write our pre-made image to a micro-SD card and run a setup script (easier).
Write our Pre-made Image to a Micro-SD Card (Recommended):

1. Write our Raspberry Pi image to an 8GB+ micro-SD card:
    a. (On Windows) - Install Win32DiskImager.
    b. Download and unzip our Raspberry Pi Image from Box.
    c. Connect your micro-SD card to your computer.
    d. Select your disk in Win32DiskImager.
    e. Open the file-selector in Win32DiskImager and choose the downloaded Raspberry Pi image.
    f. Select ‘Write’ in Win32DiskImager and wait for it to complete.
2. Starting up the Raspberry Pi:
    a. Plug in the micro-SD card into the Raspberry Pi and boot it up.
    b. Open a terminal and type:

    ```
    ./startap.sh
    ```

    c. Open a new terminal and type:

    ```
    ./start_node.sh
    ```

Flash the Raspberry Pi Yourself:

1. Flash the “Raspbian Stretch with Desktop” image onto a Raspberry Pi: https://www.raspberrypi.org/downloads/raspbian/
    a. To do this, download the zip file: 2018-10-09-raspbian-stretch.zip
    b. Flash the image onto the Raspberry Pi’s SD card using Etcher: https://etcher.io

2. Boot up the Raspberry Pi and walk through the setup steps. Be sure to connect to wifi.
    a. Skip the software update
    b. After finishing the setup steps, open up Chromium and try to search for something
    c. You may need to enter in your credentials to connect to your LAN
3. Clone the Brypt Node Git repository from Github.com. Do this on both your client computer and designated Raspberry PI coordinator.
    a. *cd ~*
    b. *git clone https://github.com/Stygain/brypt-node.git*
4. General purpose computer install (Provided VM).
    a. Ensure ZeroMQ and OpenSSL are installed on your system from *Part One.*
    b. Open a Command Line Interface (CLI) window to your local copy of brypt-node.
    c. Move to the dev folder and run the make command.
        i. *cd dev*
        ii. *make*
5. Raspberry PI install
    a. Open a Command Line Interface (CLI) to your local copy of brypt-node.
        i. *cd brypt-node*
    b. Move to the dev folder, run the setup script, and build the binary.
        i. *cd dev*
        ii. *chmod +x install.sh*
        iii. *sudo ./install.sh*
            1. *Type 'y' as prompted. Hit 'q' or 'Ok' to continue through any other prompts.*
        iv. *After reboot, open a new CLI:*
            1. *cd brypt-node/dev*
        v. *make*
6. Starting a root coordinator on the Raspberry PI
    a. Once the device has rebooted open a new CLI window and ensure the Access Point is running.
        i. *sudo service hostapd status*
            1. **If hostapd is not running**:
                a. *sudo hostapd /etc/hostapd/hostapd.conf*
        ii. *sudo service dnsmasq status*
            1. **If dnsmasq is not running**: In the dev folder of brypt-node run the Access Point startup script and reboot the device.
                a. *cd ./brypt-node/dev/config/AP*
                b. *sudo ./startup_ap.sh*
                c. If startup_ap.sh fails to run, make sure:
                    i. *perl -pe 's/\r$//' < startup_ap.sh > startup_ap.sh2*
                    ii. *rm -f ./startup_ap.sh*
                    iii. *mv ./startup_ap.sh2 ./startup_ap.sh*
                    iv. *chmod +x startup_ap.sh*

            v. *sudo ./startup_ap.sh*
         d. The device should reboot automatically, but if not:
            i. *sudo reboot*
   b. Open a new CLI window to dev folder of brypt-node and start the coordinator.
      i. *cd brypt-node/dev*
      ii. *git checkout multi-punch*
      iii. *make*
      iv. *make root*
      v. Troubleshooting: If you get a memory corruption error, run:
         1. *make clean*
         2. *make*

7. (Optional) To connect a leaf node to the Brypt root coordinator, use a general purpose computer running MacOS:
   a. Since the node must be in the coordinator's access point you must connect into the network after the desktop connection has been logged in. Otherwise the application will not be able to query the hosted central server at https://www.brypt.com.
   b. To add a leaf node, repeat steps 3-5 but **do not choose to start up the Access Point on reboot** when prompted.
   c. Open the brypt-node/dev folder and start the leaf.
      i. *make*
      ii. *make leaf_two*

(Optional) Part Three: Brypt Network - Resource Constrained Nodes

1. Install required libraries and resources.
   a. IDE
      i. https://www.arduino.cc/en/main/software
   b. Arduino IDE library for your Adafruit Feather
      i. https://www.adafruit.com/category/943
   c. Install Crypto library by Rhys Weatherley (Sketch -> Include library -> Manage libraries)
   d. Follow Installation instructions for arduino-LoRa library: https://github.com/sandeepmistry/arduino-LoRa
2. Ensure you have the brypt-node repository cloned from *Part Two.*
3. Create a new folder called 'Message' in your Arduino IDE installation library folder. (For Windows: Program Files (x86)/Arduino/libraries)
4. Copy brypt-node/arduino/message/message.hpp and brypt-node/arduino/message/utility.hpp in the newly created 'Message' folder.

5. Open the Arduino.ino file in the Arduino IDE for your desired device (.ino files are found under brypt-node/arduino/SPECIFIC_DEVICE/).
6. Ensure the coordinator is running as this is a proof concept and the leaf node must be able make initial contact once it is flashed.
7. Flash the .ino file onto the Adafruit Feather using the Arduino IDE (Click right facing arrow 'Upload').

(Optional) Part Four: Brypt Server

1. Install the Go programming language to your system.
    a. https://golang.org/doc/install
2. Clone the Brypt Server Git repository from Github.com into your Go *src* path.
    a. *git clone https://github.com/Stygain/brypt-server.git*
3. Open a Command Line Interface (CLI) to your local copy of brypt-server (i.e. *$GOPATH/src/brypt-server)*.
4. Get and install the Brypt Server Go dependencies.
    a. *make deps*
    b. *make add-deps*
5. Build the Brypt Server binary.
    a. *make build*
6. Run the Brypt Server binary.
    a. *./bin/bserv*
7. If desired, the brypt-server *production* branch may be used on Heroku to create a hosted server.

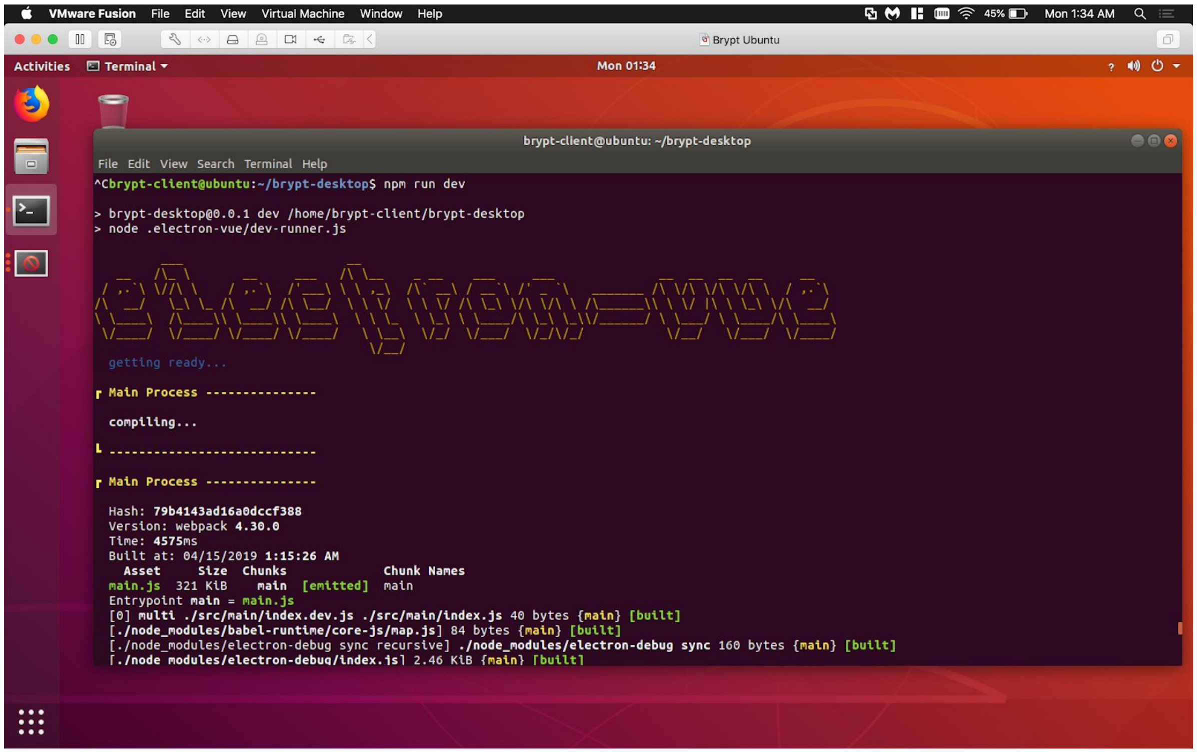
Part Five: Demo Walkthrough and Network Lifecycle

1. Ensure that each application and network component is fully installed.
2. Startup the Brypt network root coordinator. This will be the Raspberry PI 3 running the local Access Point.
    a. To easily verify the AP is running view your local Wifi connections and a *brypt-net-000000* ssid should be visible.
    b. To verify that an IPv4 address is assigned to connected devices, connect to the AP and run *ifconfig*. From *ifconfig* your Wifi interface should have some address starting with 192.168.30.x.
3. If testing with an additional Raspberry Pi 3 or additional Adafruit Feather nodes, you can connect them now.
4. With the Brypt network running start the desktop application on the client computer.
    a. If you are still connected to the Brypt AP, disconnect and connect to a Wifi AP with direct internet access.
5. Login to the Brypt network through the application interface

    a. This process will authenticate your user account and provide the program with any requisite network information (i.e. the Brypt AP ssid).
    b. You will be provided a success message and the login window will be closed to open the network dashboard.
6. The dashboard interface will display a processing animation while it searches for the Brypt AP.
    a. If a notification is presented that it has found the network, click the button to connect.
    b. If a notification is presented that a network has not been found:
        i. Click the button to re-scan the local networks
        ii. If the issue persists, ensure the Brypt root coordinator is hosting an Access Point and assigning devices a proper IPv4 address.
7. Once connected into the Brypt network, the dashboard interface will provide information on the connected nodes, arbitrary “sensor” readings, and additional network data processing modules.
8. A cycle reading request is sent every thirty seconds and the aggregated data will be placed into the streaming chart.

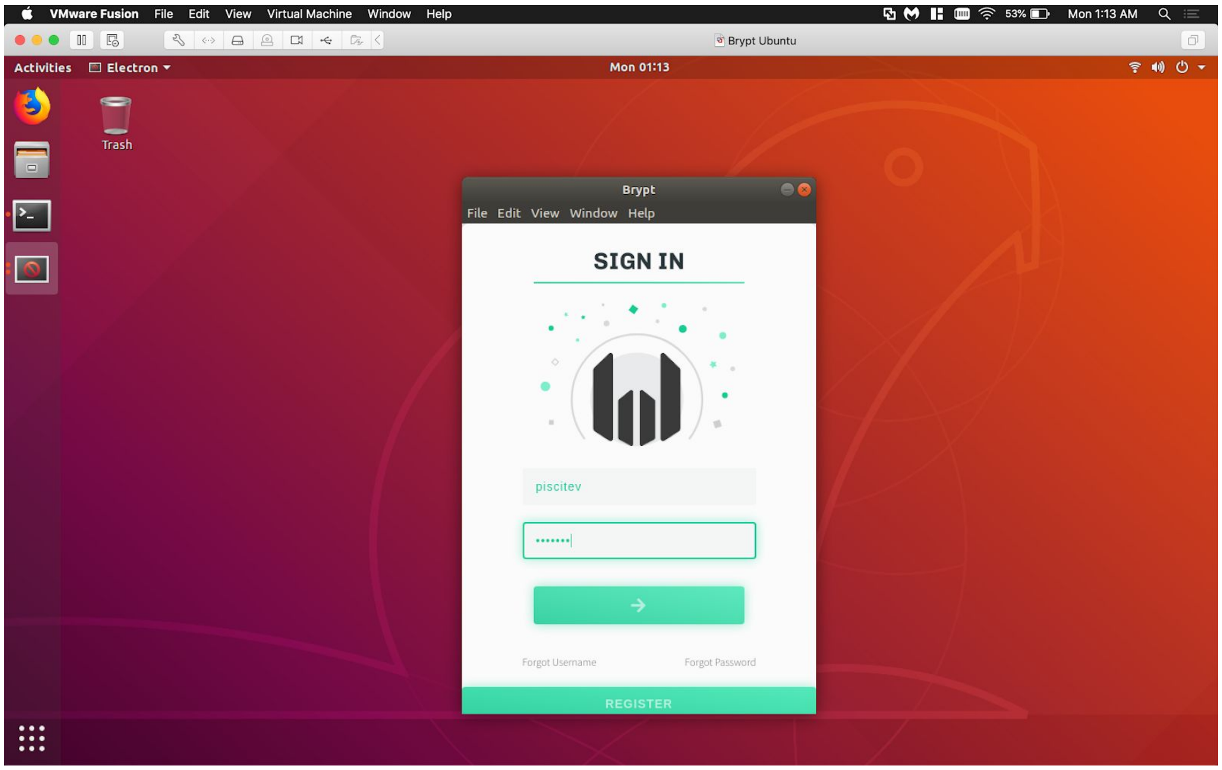
Part Six: Brypt Ubuntu Virtual Machine in Action

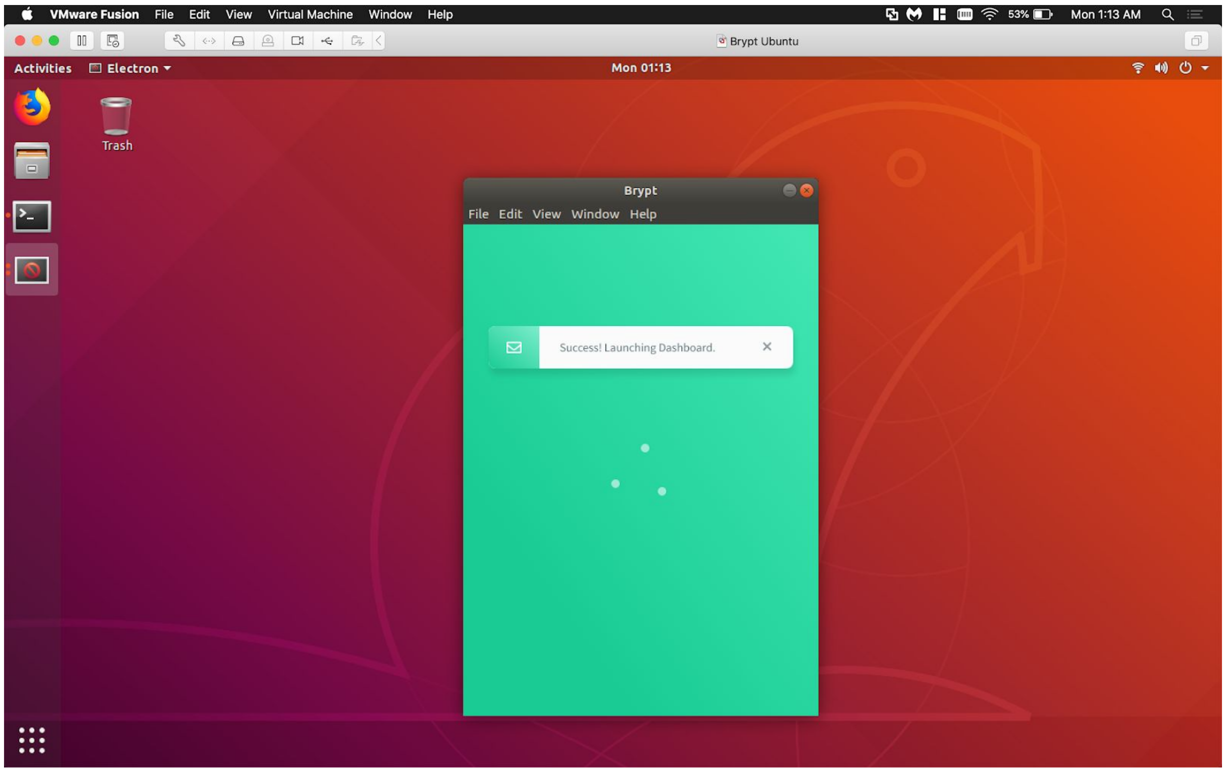
1. Starting up Brypt Desktop



```
^Cbrypt-client@ubuntu:~/brypt-desktop$ npm run dev

> brypt-desktop@0.0.1 dev /home/brypt-client/brypt-desktop
> node .electron-vue/dev-runner.js

  getting ready...

┏ Main Process ---------------

  compiling...

┗ ---------------------------

┏ Main Process ---------------

  Hash: 79b4143ad16a0dccf388
  Version: webpack 4.30.0
  Time: 4575ms
  Built at: 04/15/2019 1:15:26 AM
    Asset      Size  Chunks             Chunk Names
  main.js  321 KiB    main  [emitted]  main
  Entrypoint main = main.js
  [0] multi ./src/main/index.dev.js ./src/main/index.js 40 bytes {main} [built]
  [./node_modules/babel-runtime/core-js/map.js] 84 bytes {main} [built]
  [./node_modules/electron-debug sync recursive] ./node_modules/electron-debug sync 160 bytes {main} [built]
  [./node_modules/electron-debug/index.js] 2.46 KiB {main} [built]
```

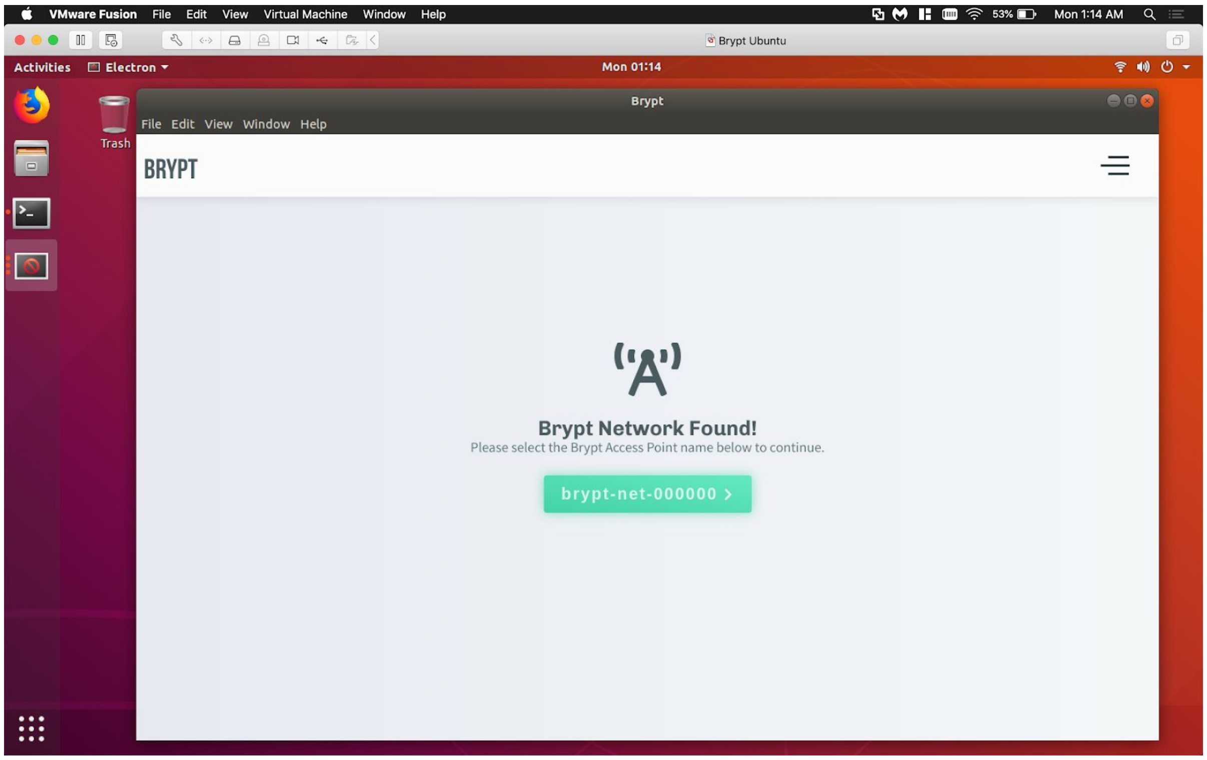
2. Logging in to the Brypt Application




VMware Fusion File Edit View Virtual Machine Window Help
Mon 1:13 AM
Brypt Ubuntu
Activities Electron
Mon 01:13
Trash
Brypt
File Edit View Window Help

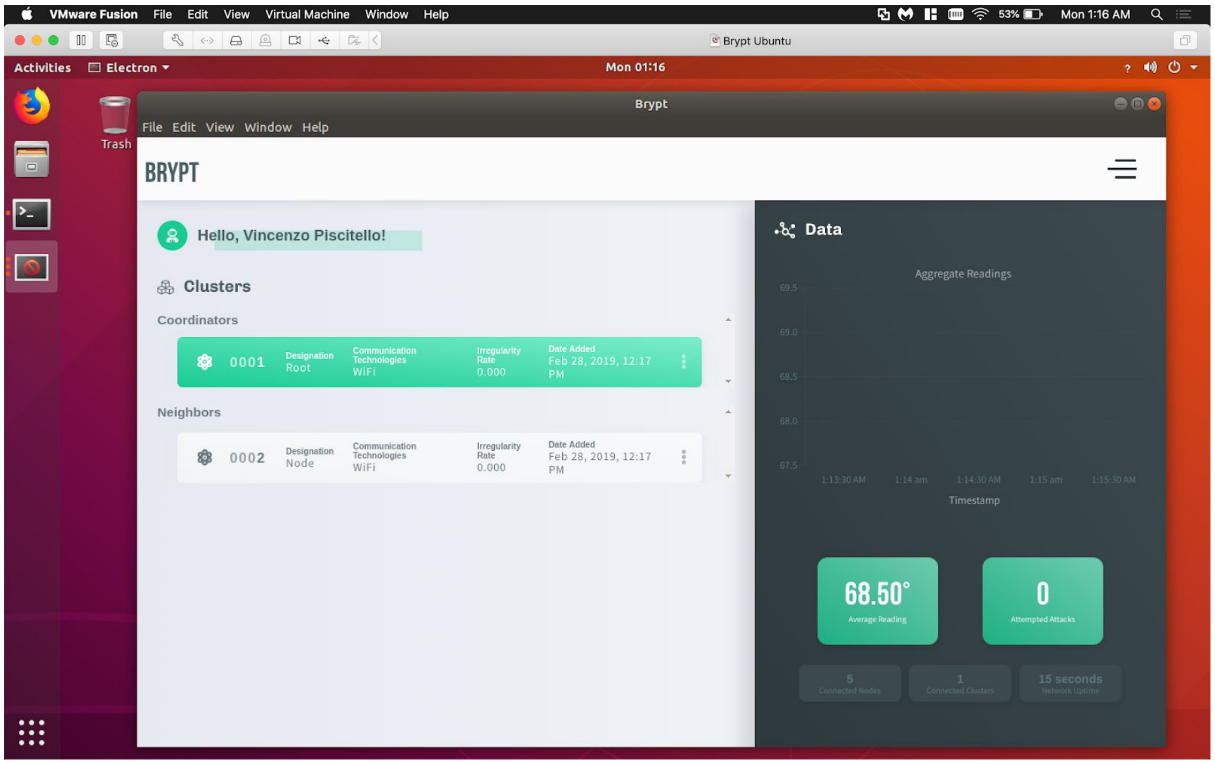
Success! Launching Dashboard.

3. Finding the Brypt Network


VMware Fusion File Edit View Virtual Machine Window Help
53% Mon 1:14 AM
Brypt Ubuntu
Activities Electron
Mon 01:14
Trash
Brypt
File Edit View Window Help
BRYPT

Brypt Network Found!
Please select the Brypt Access Point name below to continue.

4. Initial Brypt Dashboard with one root coordinator and one leaf node


VMware Fusion File Edit View Virtual Machine Window Help
53% Mon 1:16 AM
Brypt Ubuntu
Activities Electron
Mon 01:16
Trash
Brypt
File Edit View Window Help
BRYPT

Hello, Vincenzo Piscitello!

Clusters

Coordinators

	Designation	Communication Technologies	Irregularity Rate	Date Added
0001	Root	WiFi	0.000	Feb 28, 2019, 12:17 PM

Neighbors

	Designation	Communication Technologies	Irregularity Rate	Date Added
0002	Node	WiFi	0.000	Feb 28, 2019, 12:17 PM

68.50° Average Reading

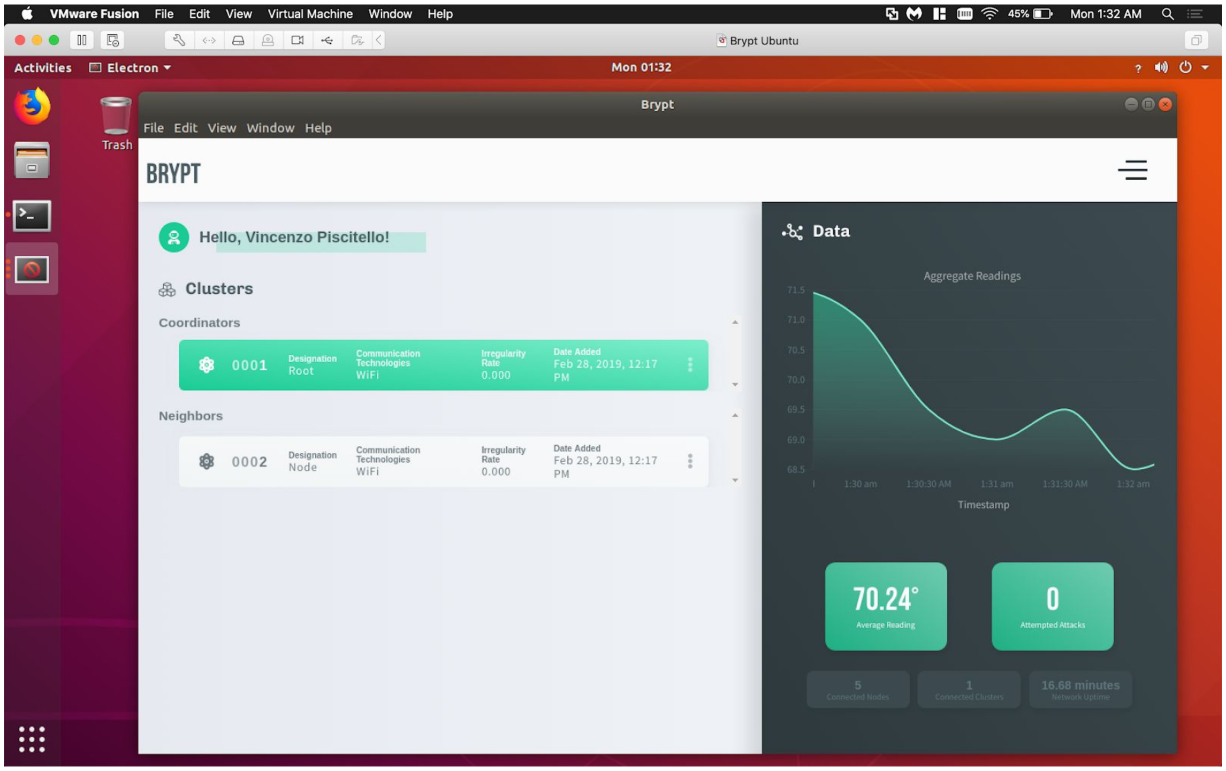
0 Attempted Attacks

5 Connected Nodes

1 Connected Clusters

15 seconds Network Uptime

5. Brypt Dashboard and readings collected after 15 minutes of running



Hello, Vincenzo Piscitello!

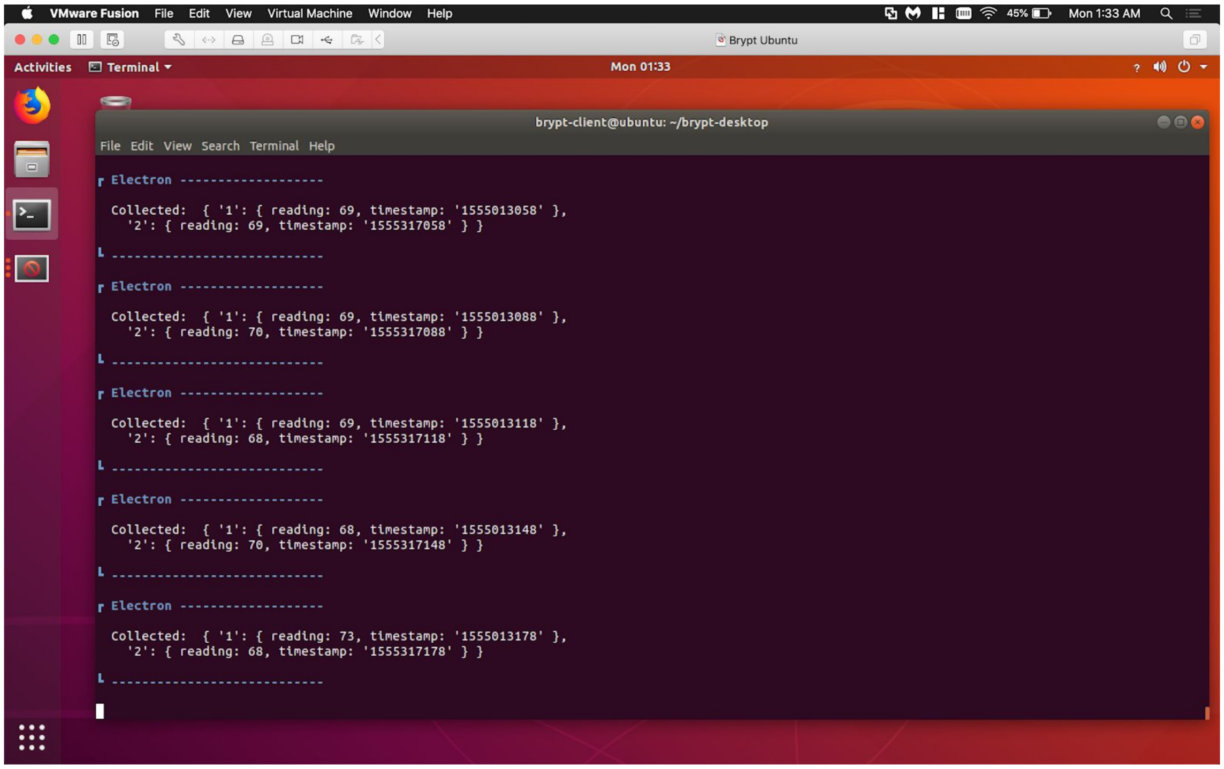
Clusters

Coordinators

	Designation	Communication Technologies	Irregularity Rate	Date Added
0001	Root	WiFi	0.000	Feb 28, 2019, 12:17 PM

Neighbors

	Designation	Communication Technologies	Irregularity Rate	Date Added
0002	Node	WiFi	0.000	Feb 28, 2019, 12:17 PM


VMware Fusion File Edit View Virtual Machine Window Help
Mon 1:33 AM
Brypt Ubuntu
Activities
Terminal
Mon 01:33
brypt-client@ubuntu: ~/brypt-desktop
File Edit View Search Terminal Help

```
┌ Electron ------------------

  Collected:  { '1': { reading: 69, timestamp: '1555013058' },
    '2': { reading: 69, timestamp: '1555317058' } }

└ --------------------------

┌ Electron ------------------

  Collected:  { '1': { reading: 69, timestamp: '1555013088' },
    '2': { reading: 70, timestamp: '1555317088' } }

└ --------------------------

┌ Electron ------------------

  Collected:  { '1': { reading: 69, timestamp: '1555013118' },
    '2': { reading: 68, timestamp: '1555317118' } }

└ --------------------------

┌ Electron ------------------

  Collected:  { '1': { reading: 68, timestamp: '1555013148' },
    '2': { reading: 70, timestamp: '1555317148' } }

└ --------------------------

┌ Electron ------------------

  Collected:  { '1': { reading: 73, timestamp: '1555013178' },
    '2': { reading: 68, timestamp: '1555317178' } }

└ --------------------------
```