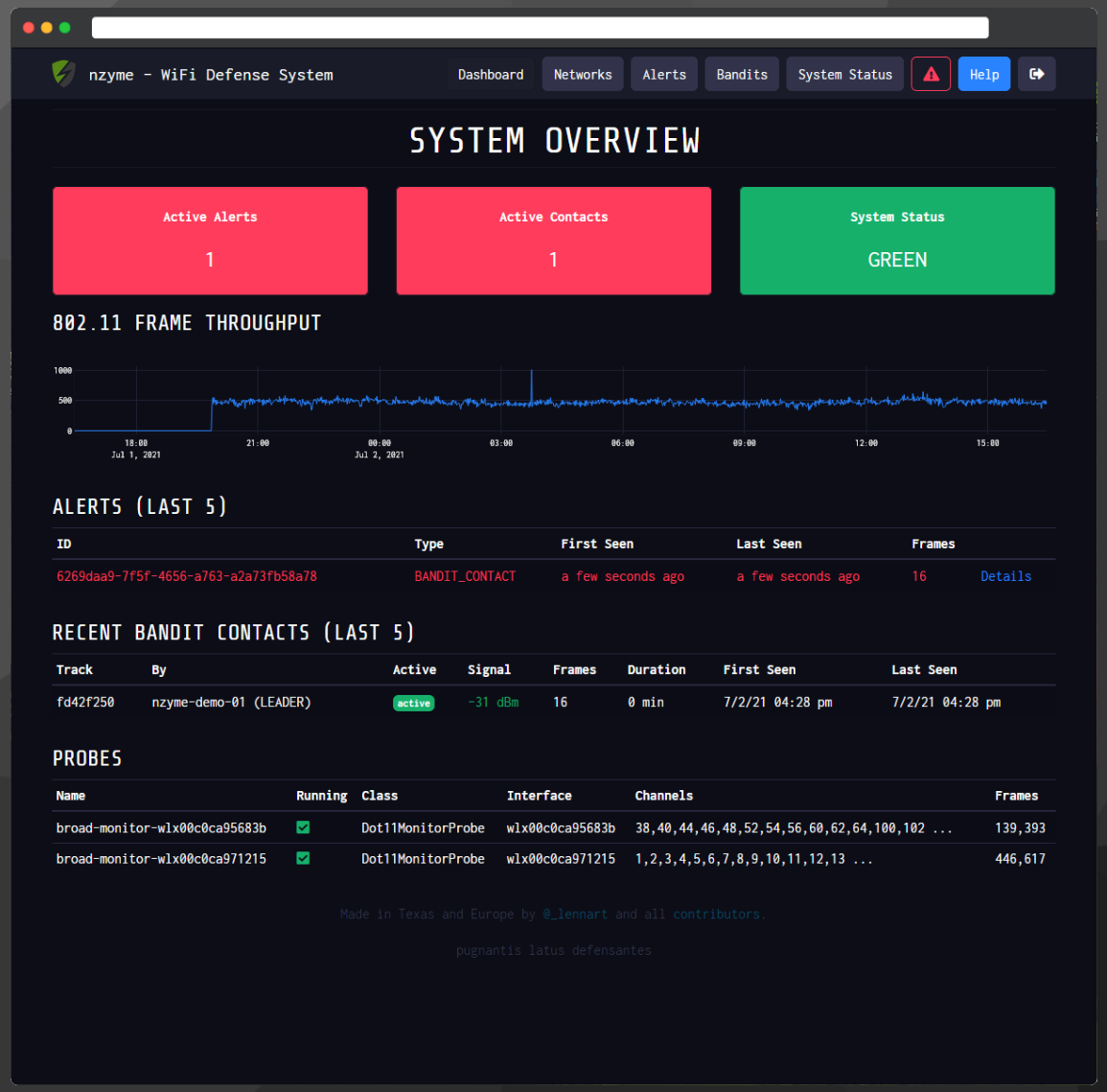


Nzyme Technical Segment

Paul Asadoorian & Larry Pesce

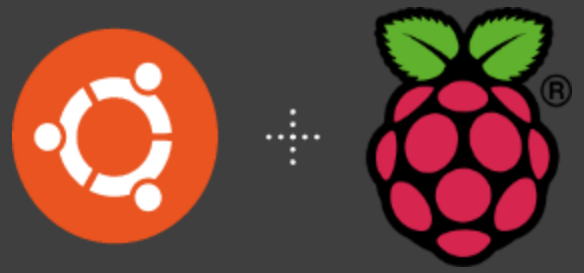
Security Weekly

Hardware

- Paul's Setup:
  - Panda Wireless PAU09 N600 https://www.amazon.com/gp/product/B01LY35HGO/ ($42.99)
  - Vilros Raspberry Pi 4 8GB Basic Starter Kit with Fan Cooled Heavy Duty Aluminum Alloy Case https://www.amazon.com/gp/product/B089ZZ6DN5/ ($109.99)

- Larry's Setup
  - Panda PAU09 N600 https://www.amazon.com/gp/product/B01LY35HGO/ ($42.99)
  - Vilros Pi 2 2 Gig kit https://www.amazon.com/gp/product/B07XTRFD3Z/ ($89.99)

Software



- Paul used Ubuntu for Raspberry PI (https://ubuntu.com/raspberry-pi)
  - Gives you Ubuntu 20.04 server running on the PI

- Larry used raspios Buster 2021-05-07 (https://www.raspberrypi.org/%20downloads/)
  - Standard build for the PI

Nzyme Setup and Installation

```
$ sudo hostnamectl set-hostname nikon

$ cat 00-static-ip.yaml
network:
    version: 2
    ethernets:
        eth0:
            dhcp4: false
            addresses: [10.10.1.131/24]
            gateway4: 10.10.1.1
            nameservers:
              search: [int.psw.io]
              addresses: [10.10.1.111, 10.10.1.112]

$ cat /etc/hosts
127.0.0.1 localhost
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

127.0.1.1 nikon

$ sudo netplan apply
```

```
$ sudo apt update && apt upgrade

$ sudo apt install -y libpcap0.8 openjdk-11-jre-headless
postgresql-12 wireless-tools

$ wget https://assets.nzyme.org/releases/nzyme-1.1.1.deb

$ sudo dpkg -i nzyme-1.1.1.deb

$ sudo -u postgres psql
postgres=# create database nzyme;
CREATE DATABASE
postgres=# create user nzyme with encrypted password
'YOUR_PASSWORD_HERE';
CREATE ROLE
postgres=# grant all privileges on database nzyme to nzyme;GRANT
postgres=# \q
```

Configure The Wireless Interface

```
# Set the interface to monitor mode
$ ip link set wlan0 down

$ iw wlan0 set monitor none

$ ip link set wlan0 up

# Make sure we are actually in monitor mode
$ iw dev
phy#8
    Interface wlan0
        ifindex 11
        wdev 0x800000001
        addr 9c:ef:d5:fa:ac:fa
        type monitor
        channel 44 (5220 MHz), width: 20 MHz (no HT), center1: 5220 MHz
        txpower 20.00 dBm

# Get a comma separated list of the supported channels for the configuration file
$ iwlist wlan0 channel | grep 'Channel' | grep -v 'Current' | awk '{print $2}' | tr '\n' ','
01,02,03,04,05,06,07,08,09,10,11,12,13,14,36,38,40,42,44,46,48,52,54,56,58,60,62,64,100,102,104,106,

# Generate a password hash for the admin interface (save this, you'll need it later)
$ echo -n secretpassword | sha256sum
```

Configuration - /etc/nzyme/nzyme.conf

```
general: {
  role: LEADER

  # The ID or name of this nzyme instance. Must be unique and contain only alphanumeric characters, underscores and dashes.
  id: nzyme-node-01

  # You will use this password to log in to the web interface. (From previous step)
  admin_password_hash: 3d2a100dcce86f8539b0cf717fedd4e428e64d17624b3a9e77524da816f6220c

  # Path to postgreSQL database. Make suer to change username, password and database name. (This is described in the documentation)
  # NOTE: Use the creds from the previous step
  database_path: "postgresql://localhost:5432/nzyme?user=nzyme&password=secret"

  # Download current list of manufacturers and enable MAC address to manufacturer lookup?
  fetch_ouis: true

  # Path to directory that the tracker will use to store some temporary information. (must be writable)
  data_directory: /usr/share/nzyme

  # We use Python to inject frames for traps.
  # NOTE: Larry had to change this to 3.7 on raspbian
  python {
    # Path to python executable. (nzyme supports both Python 3 and 2)
    path: /usr/bin/python3.8

    # Script directory. This must be an existing and writable directory. We'll store some generated Python scripts here.
    script_directory: /tmp

    # Script prefix. A prefix for the generate scripts. There is usually no reason to change this setting.
    script_prefix: nzyme_
  }
```

Setup The Web Interface

```
# Web interface and REST API configuration.
interfaces: {
  # Make sure to set this to an IP address you can reach from your workstation.
  rest_listen_uri: "http://10.10.1.131:22900/"

  # This is usually the same as the `rest_listen_uri`. Take a look at the configuration documentation to
learn about
  # other use-cases. It will be interesting if you run behind a load balancer or NAT. (basically, it is
the address
  # that your web browser will use to try to connect to nzyme and it has to be reachable for it.)
  http_external_uri: "http://10.10.1.131:22900/"

  # Use TLS? (HTTPS) See https://go.nzyme.org/docs-https
  use_tls: false
}
```

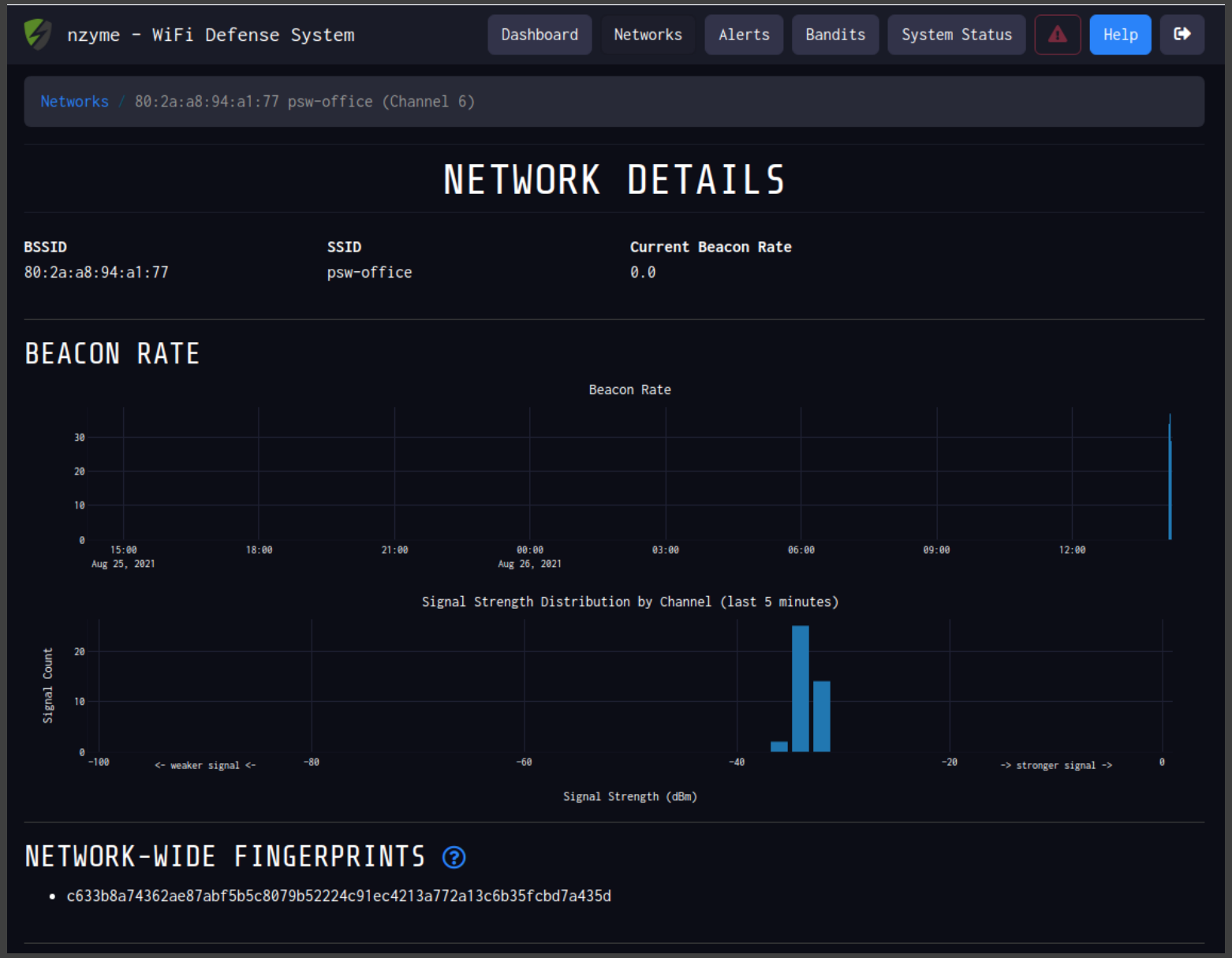
Configure 802.11 Monitors

```
802_11_monitors: [
  {
    # The 802.11/WiFi adapter name. (from `ifconfig` or `ip link`)
    device: wlan0


    # WiFi interface and 802.11 channels to use. Nzyme will cycle your network adapters through these channels.
    # Consider local legal requirements and regulations.
    # See also: https://en.wikipedia.org/wiki/List_of_WLAN_channels
    #channels: [1,2,3,4,5,6,7,8,9,10,11]
    channels: [1,2,3,4,5,6,7,8,9,10,11,12,13,14,36,38,40,42,44,46,48,52,54,56,58,60,62,64,100,102,104,106]


    # There is no way for nzyme to configure your wifi interface directly. We are using direct operating system commands to
    # configure the adapter. Examples for Linux are in the documentation.
    channel_hop_command: "sudo /sbin/iwconfig {interface} channel {channel}"


    # Channel hop interval in seconds. Leave at default if you don't know what this is.
    channel_hop_interval: 1


    # Skip the automatic monitor mode configuration of this interface. Only enable this if for some reason libpcap can't
    # properly configure this interface into monitor mode. In that case, you can try to set it manually instead.
    skip_enable_monitor: true
  }
]
```

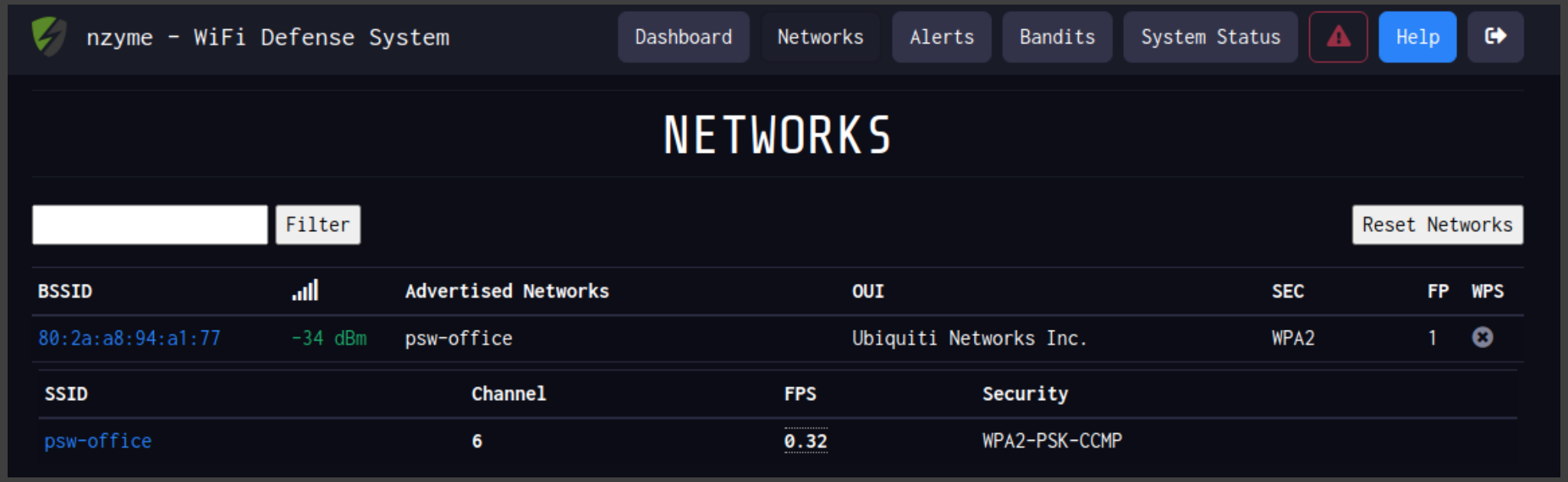
Define Your SSIDs and BSSIDs

```
{
    ssid: YOUR-SSID-HERE

    # The list from the previous command
    channels: [1,2,3,4,5,6,7,8,9,10,11,12,13,14,36,38,40,42,44,46,48,52,54,56,58,60,62,64,100,102,104,106]
    security: [WPA2-PSK-CCMP]

    # You may have to adjust this:
    beacon_rate: 40

    # In the web interface, copy and paste the address and the fingerprints
    # Your APs may have more than one fingerprint for the radios (one for 2.4GHz and one for 5GHz)
    bssids: [
      {
        address: "xx:xx:xx:xx:xx:xx",
        fingerprints: [
          c633b8a74362ae87abf5b5c8079b52224c91ec4213a772a13c6b35fcbd7a435d
          3693fc1122eeb54107b21f4143ea73b527909ae07d8d6f83c0c318657c40cead
        ]
      }
     {
        address: "xx:xx:xx:xx:xx:xx",
        fingerprints: [
          c633b8a74362ae87abf5b5c8079b52224c91ec4213a772a13c6b35fcbd7a435d
          3693fc1122eeb54107b21f4143ea73b527909ae07d8d6f83c0c318657c40cead
        ]
     }
```

Network Details



WIFI Networks



Logs - /var/log/nzyme/nzyme.log

```
Aug 26 17:47:25 nikon nzyme[2698]: Aug 26, 2021 5:47:25 PM liquibase.lockservice
Aug 26 17:47:25 nikon nzyme[2698]: INFO: Successfully released change log lock
Aug 26 17:47:43 nikon nzyme[2698]: Can't restore interface wlan1 wireless mode (SIOCSIWMODE failed: Bad file descriptor).
Aug 26 17:47:43 nikon nzyme[2698]: Please adjust manually.
```

I looked here because my card kept disappearing from nzyme...no errors in the logs.

Turns out after examining the output of dmesg that my WIFI adapter kept generating these messages:

ieee80211 phy5: rt2x00usb_vendor_request: Error - Vendor Request 0x07 failed for offset 0x101c with error -110

Turns out that is related to power! I had the Panda PAU09 plugged into a USB3 port on the Pi, and I remembered back that there are still some compatibility issues with USB3 on the Pi; I remember reading a while back (and experienced) that a keyboard plugged into USB3 was unstable/unusable. Swapped the WIFI adapter to a USB2 port and it has been rock solid ever since.

References

https://www.linuxscrew.com/raspberry-pi-static-ip

https://www.nzyme.org/docs/installation-configuration/install-raspberry-pi-os