Candela TECHNOLOGIES
Network Testing and Emulation Solutions

sales@candelatech.com
support@candelatech.com
+1 (360) 380-1618 [PST, GMT -8]

Automated scanning of SSID, BSSID, and Signal of available wireless APs

Goal: Create a station and scan for SSID, BSSID, and Signal of available wireless APs
We will learn how to use a script to create a station and scan for available APs. We will then look at the /scanresults/ URI and the info we can get from a scan through JSON. Please refer to sta_scan_test.py as an example script.

1. Using the Script

A. Command Line Options

```
--sta_name nameOfStation
```

Specifies the name of the station to be created, if this option is used, the name will default to **sta0000**.

```
--ssid nameOfNetwork
```

Specifies the name of the network to connect to.
This value must be used, however, the SSID does not have to exist and a fake name can be used.

```
--security {WEP, WPA, WPA2, WPA3, Open}
```

Specifies the security type of the network to connect to.
This value must be used, however, if a fake SSID is used the type should be open.

B. Running the script

A. As an example, we can run the script using:

```
./sta_scan_test.py --sta_name sta0000 --ssid fake_ssid --security open --radio 
```

B. This will produce output that looks like this:

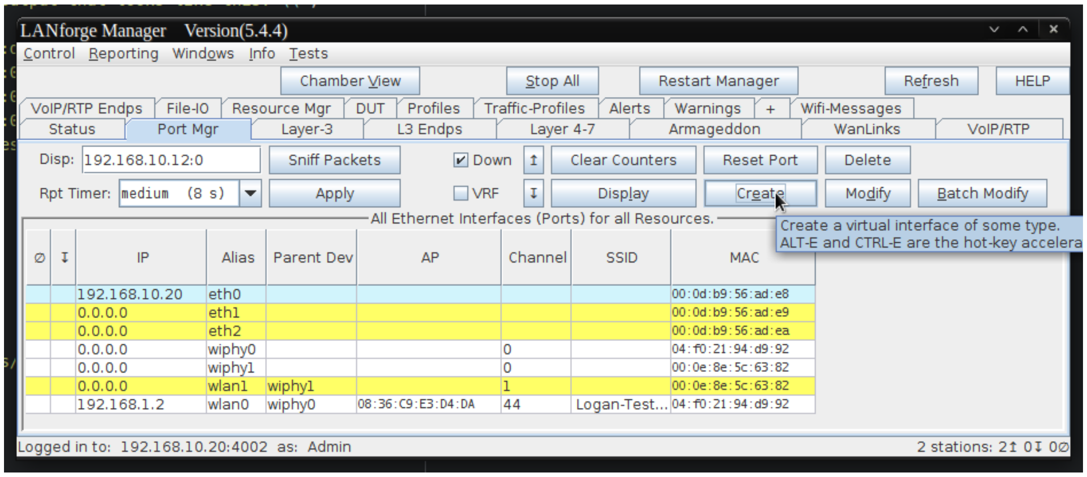
```
BSS                      Signal  SSID
08:36:c9:e3:d4:da        -32.0   Logan-Test-Net
10:56:11:0c:04:02        -80.0   :)
22:56:11:0c:04:02        -79.0   xfinitywifi
32:56:11:0c:04:02        -80.0   NA
```

This script produces limited output, for more detail we can look at the webpage hosted by LANforge.

2. The /scanresults/ URI

A. In order to view this page we will need to create a station and start a scan.

A. First we will create the station (Make sure to click on a radio in the Port Mgr tab first):


LANforge Manager Version(5.4.4)
Control Reporting Windows Info Tests
Chamber View
Stop All
Restart Manager
Refresh
HELP
VoIP/RTP Endps File-IO Resource Mgr DUT Profiles Traffic-Profiles Alerts Warnings + Wifi-Messages
Status Port Mgr Layer-3 L3 Endps Layer 4-7 Armageddon WanLinks VoIP/RTP
Sniff Packets
Clear Counters
Reset Port
Delete
Apply
Display
Create
Modify
Batch Modify

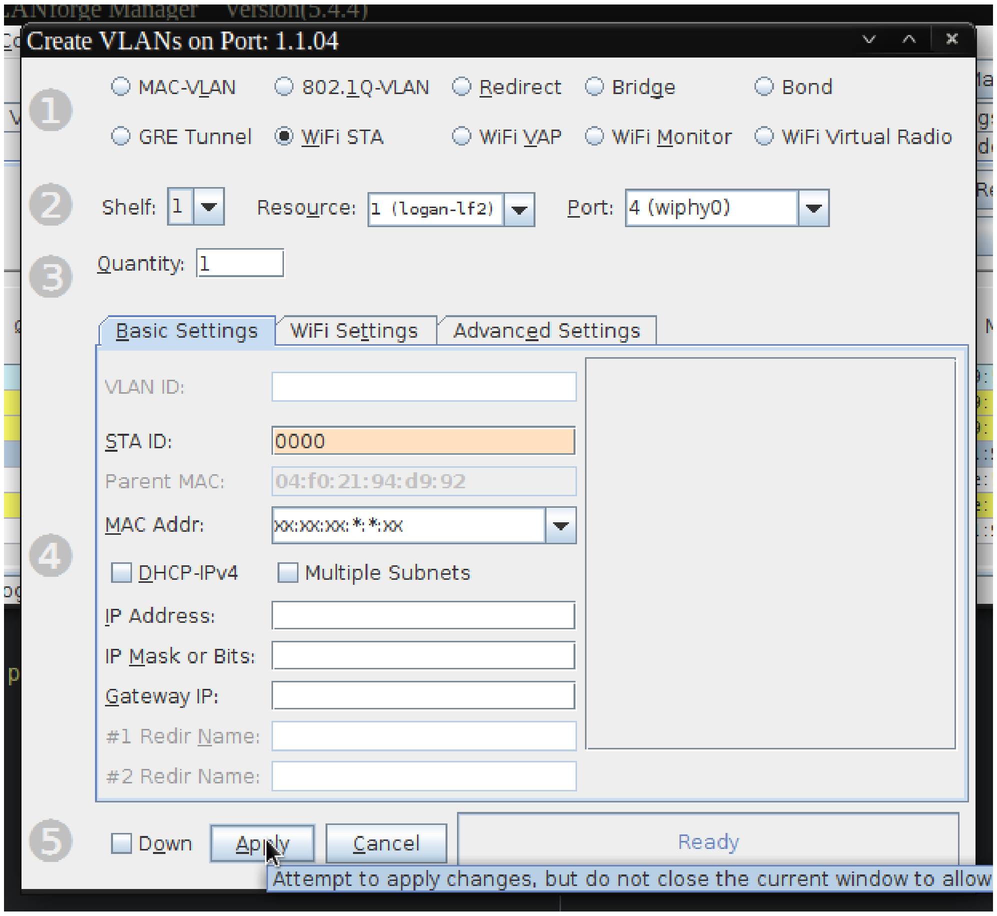
Disp: 192.168.10.12:0
Rpt Timer: medium (8 s)
✓ Down
VRF
Create a virtual interface of some type.
ALT-E and CTRL-E are the hot-key accelera

All Ethernet Interfaces (Ports) for all Resources.

∅	↧	IP	Alias	Parent Dev	AP	Channel	SSID	MAC
		192.168.10.20	eth0					00:0d:b9:56:ad:e8
		0.0.0.0	eth1					00:0d:b9:56:ad:e9
		0.0.0.0	eth2					00:0d:b9:56:ad:ea
		0.0.0.0	wiphy0			0		04:f0:21:94:d9:92
		0.0.0.0	wiphy1			0		00:0e:8e:5c:63:82
		0.0.0.0	wlan1	wiphy1		1		00:0e:8e:5c:63:82
		192.168.1.2	wlan0	wiphy0	08:36:C9:E3:D4:DA	44	Logan-Test...	04:f0:21:94:d9:92

Logged in to: 192.168.10.20:4002 as: Admin
2 stations: 2↥ 0↧ 0∅

B. Next we will create the station, the default values can be used or a specific number for the station can be given:



Create VLANs on Port: 1.1.04

1 ◯ MAC-VLAN ◯ 802.1Q-VLAN ◯ Redirect ◯ Bridge ◯ Bond ◯ GRE Tunnel ● WiFi STA ◯ WiFi VAP ◯ WiFi Monitor ◯ WiFi Virtual Radio

2 Shelf: 1 Resource: 1 (logan-lf2) Port: 4 (wiphy0)

3 Quantity: 1

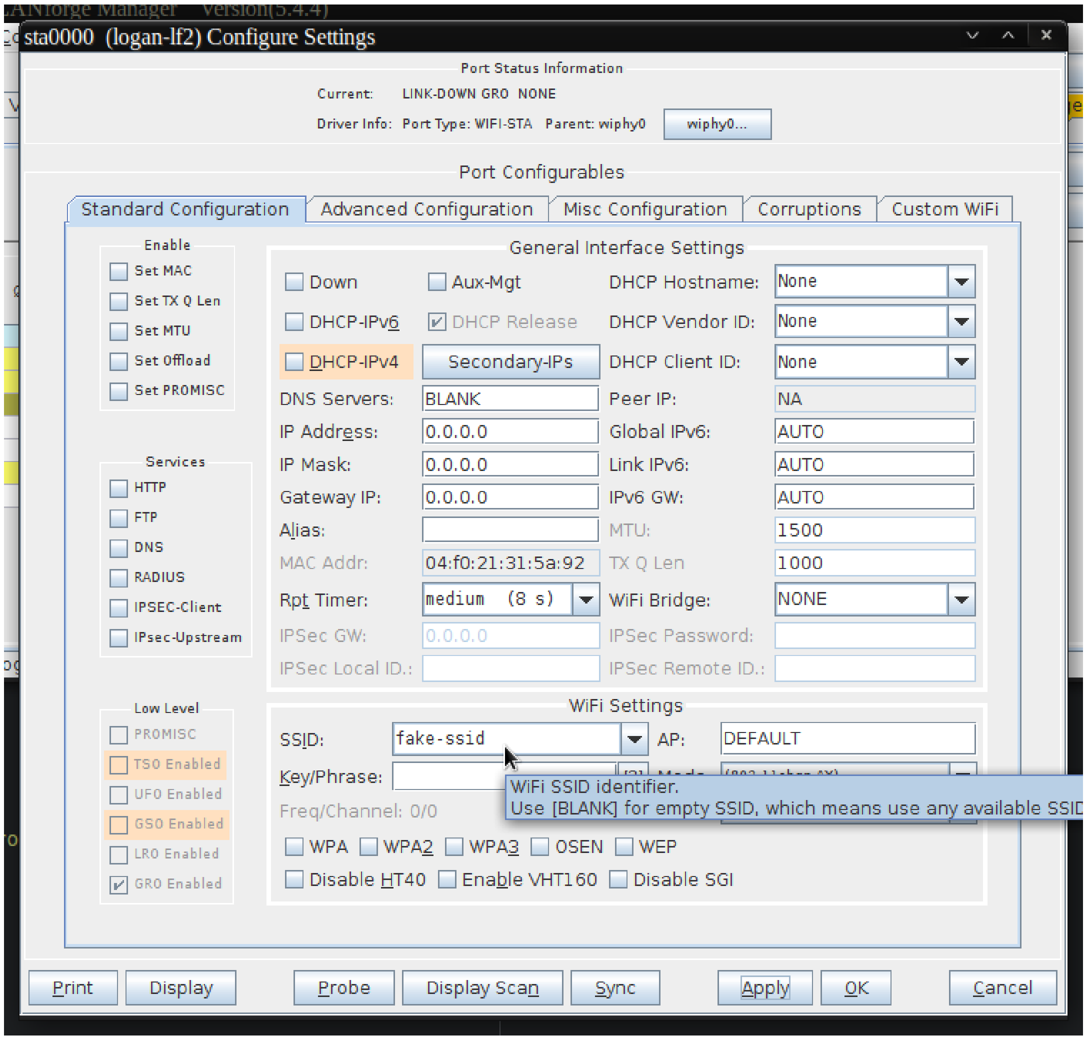
4 Basic Settings | WiFi Settings | Advanced Settings

VLAN ID:
STA ID: 0000
Parent MAC: 04:f0:21:94:d9:92
MAC Addr: xx:xx:xx:*:*:xx
☐ DHCP-IPv4 ☐ Multiple Subnets
IP Address:
IP Mask or Bits:
Gateway IP:
#1 Redir Name:
#2 Redir Name:

5 ☐ Down Apply Cancel Ready

Attempt to apply changes, but do not close the current window to allow

C. After creating the station, we will give the an SSID to connect to. (This doesn't have to be a real AP):



sta0000 (logan-lf2) Configure Settings

Port Status Information
Current: LINK-DOWN GRO NONE
Driver Info: Port Type: WIFI-STA Parent: wiphy0 wiphy0...

Port Configurables
Standard Configuration | Advanced Configuration | Misc Configuration | Corruptions | Custom WiFi

Enable: ☐ Set MAC ☐ Set TX Q Len ☐ Set MTU ☐ Set Offload ☐ Set PROMISC

Services: ☐ HTTP ☐ FTP ☐ DNS ☐ RADIUS ☐ IPSEC-Client ☐ IPsec-Upstream

Low Level: ☐ PROMISC ☐ TSO Enabled ☐ UFO Enabled ☐ GSO Enabled ☐ LRO Enabled ☑ GRO Enabled

General Interface Settings
☐ Down ☐ Aux-Mgt DHCP Hostname: None
☐ DHCP-IPv6 ☑ DHCP Release DHCP Vendor ID: None
☐ DHCP-IPv4 Secondary-IPs DHCP Client ID: None

DNS Servers:	BLANK	Peer IP:	NA
IP Address:	0.0.0.0	Global IPv6:	AUTO
IP Mask:	0.0.0.0	Link IPv6:	AUTO
Gateway IP:	0.0.0.0	IPv6 GW:	AUTO
Alias:		MTU:	1500
MAC Addr:	04:f0:21:31:5a:92	TX Q Len	1000
Rpt Timer:	medium (8 s)	WiFi Bridge:	NONE
IPSec GW:	0.0.0.0	IPSec Password:	
IPSec Local ID.:		IPSec Remote ID.:	

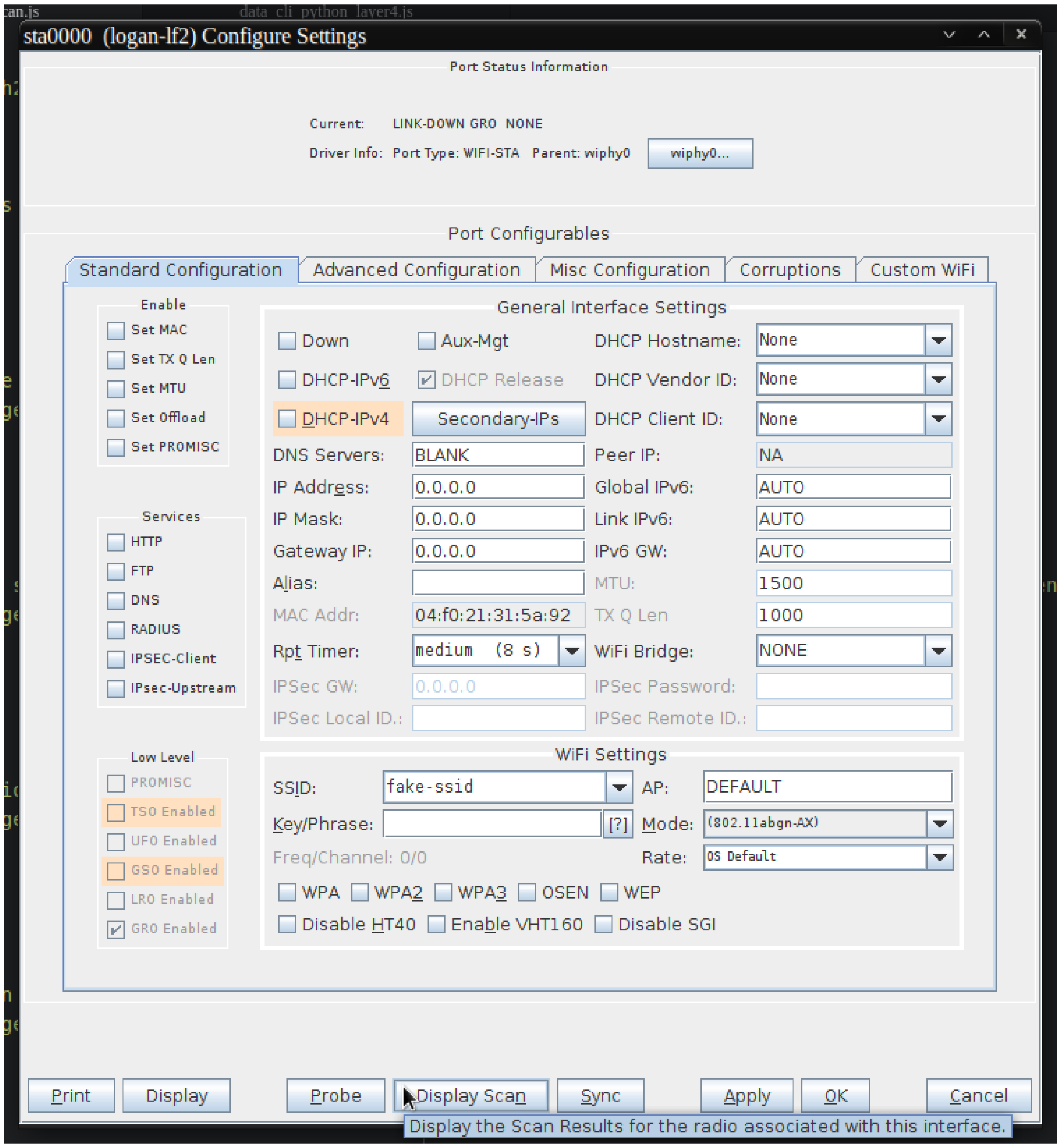
WiFi Settings
SSID: fake-ssid AP: DEFAULT
Key/Phrase:
Freq/Channel: 0/0

WiFi SSID identifier.
Use [BLANK] for empty SSID, which means use any available SSID

☐ WPA ☐ WPA2 ☐ WPA3 ☐ OSEN ☐ WEP
☐ Disable HT40 ☐ Enable VHT160 ☐ Disable SGI

Print Display Probe Display Scan Sync Apply OK Cancel

D. Clicking on Display Scan at the bottom of the station settings window will bring us to the Scan window:


sta0000 (logan-lf2) Configure Settings

Port Status Information

Current: LINK-DOWN GRO NONE

Driver Info: Port Type: WIFI-STA Parent: wiphy0

Port Configurables

DHCP Hostname:	None
DHCP Vendor ID:	None
DHCP Client ID:	None
DNS Servers:	BLANK
Peer IP:	NA
IP Address:	0.0.0.0
Global IPv6:	AUTO
IP Mask:	0.0.0.0
Link IPv6:	AUTO
Gateway IP:	0.0.0.0
IPv6 GW:	AUTO
Alias:	
MTU:	1500
MAC Addr:	04:f0:21:31:5a:92
TX Q Len	1000
Rpt Timer:	medium (8 s)
WiFi Bridge:	NONE
IPSec GW:	0.0.0.0
IPSec Password:	
IPSec Local ID.:	
IPSec Remote ID.:	

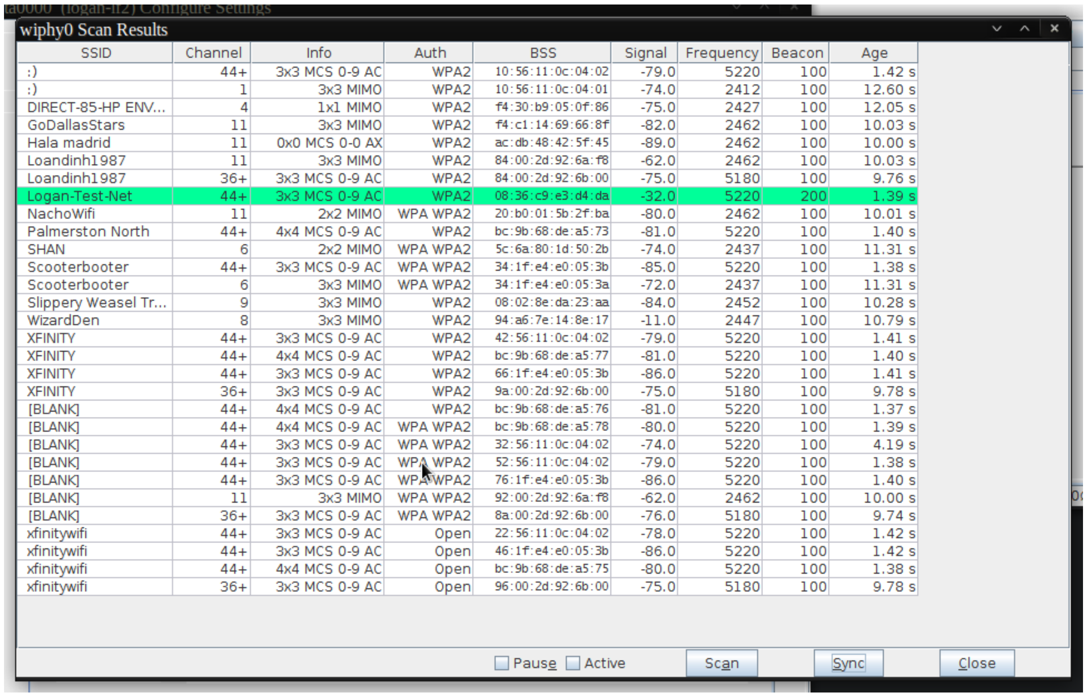
WiFi Settings

SSID:	fake-ssid
AP:	DEFAULT
Key/Phrase:	
Mode:	(802.11abgn-AX)
Freq/Channel:	0/0
Rate:	OS Default

✓ DHCP Release; ✓ GRO Enabled

Display the Scan Results for the radio associated with this interface.

E. Finally we'll be able to start the scan and see the results. Clicking on Scan and waiting a few seconds will show all of the APs availble to the station:


wiphy0 Scan Results

SSID	Channel	Info	Auth	BSS	Signal	Frequency	Beacon	Age
:)	44+	3x3 MCS 0-9 AC	WPA2	10:56:11:0c:04:02	-79.0	5220	100	1.42 s
:)	1	3x3 MIMO	WPA2	10:56:11:0c:04:01	-74.0	2412	100	12.60 s
DIRECT-85-HP ENV...	4	1x1 MIMO	WPA2	f4:30:b9:05:0f:86	-75.0	2427	100	12.05 s
GoDallasStars	11	3x3 MIMO	WPA2	f4:c1:14:69:66:8f	-82.0	2462	100	10.03 s
Hala madrid	11	0x0 MCS 0-0 AX	WPA2	ac:db:48:42:5f:45	-89.0	2462	100	10.00 s
Loandinh1987	11	3x3 MIMO	WPA2	84:00:2d:92:6a:f8	-62.0	2462	100	10.03 s
Loandinh1987	36+	3x3 MCS 0-9 AC	WPA2	84:00:2d:92:6b:00	-75.0	5180	100	9.76 s
Logan-Test-Net	44+	3x3 MCS 0-9 AC	WPA2	08:36:c9:e3:d4:da	-32.0	5220	200	1.39 s
NachoWifi	11	2x2 MIMO	WPA WPA2	20:b0:01:5b:2f:ba	-80.0	2462	100	10.01 s
Palmerston North	44+	4x4 MCS 0-9 AC	WPA2	bc:9b:68:de:a5:73	-81.0	5220	100	1.40 s
SHAN	6	2x2 MIMO	WPA WPA2	5c:6a:80:1d:50:2b	-74.0	2437	100	11.31 s
Scooterbooter	44+	3x3 MCS 0-9 AC	WPA WPA2	34:1f:e4:e0:05:3b	-85.0	5220	100	1.38 s
Scooterbooter	6	3x3 MIMO	WPA WPA2	34:1f:e4:e0:05:3a	-72.0	2437	100	11.31 s
Slippery Weasel Tr...	9	3x3 MIMO	WPA2	08:02:8e:da:23:aa	-84.0	2452	100	10.28 s
WizardDen	8	3x3 MIMO	WPA2	94:a6:7e:14:8e:17	-11.0	2447	100	10.79 s
XFINITY	44+	3x3 MCS 0-9 AC	WPA2	42:56:11:0c:04:02	-79.0	5220	100	1.41 s
XFINITY	44+	4x4 MCS 0-9 AC	WPA2	bc:9b:68:de:a5:77	-81.0	5220	100	1.40 s
XFINITY	44+	3x3 MCS 0-9 AC	WPA2	66:1f:e4:e0:05:3b	-86.0	5220	100	1.41 s
XFINITY	36+	3x3 MCS 0-9 AC	WPA2	9a:00:2d:92:6b:00	-75.0	5180	100	9.78 s
[BLANK]	44+	4x4 MCS 0-9 AC	WPA2	bc:9b:68:de:a5:76	-81.0	5220	100	1.37 s
[BLANK]	44+	4x4 MCS 0-9 AC	WPA WPA2	bc:9b:68:de:a5:78	-80.0	5220	100	1.39 s
[BLANK]	44+	3x3 MCS 0-9 AC	WPA WPA2	32:56:11:0c:04:02	-74.0	5220	100	4.19 s
[BLANK]	44+	3x3 MCS 0-9 AC	WPA WPA2	52:56:11:0c:04:02	-79.0	5220	100	1.38 s
[BLANK]	44+	3x3 MCS 0-9 AC	WPA WPA2	76:1f:e4:e0:05:3b	-86.0	5220	100	1.40 s
[BLANK]	11	3x3 MIMO	WPA WPA2	92:00:2d:92:6a:f8	-62.0	2462	100	10.00 s
[BLANK]	36+	3x3 MCS 0-9 AC	WPA WPA2	8a:00:2d:92:6b:00	-76.0	5180	100	9.74 s
xfinitywifi	44+	3x3 MCS 0-9 AC	Open	22:56:11:0c:04:02	-78.0	5220	100	1.42 s
xfinitywifi	44+	3x3 MCS 0-9 AC	Open	46:1f:e4:e0:05:3b	-86.0	5220	100	1.42 s
xfinitywifi	44+	4x4 MCS 0-9 AC	Open	bc:9b:68:de:a5:75	-80.0	5220	100	1.38 s
xfinitywifi	36+	3x3 MCS 0-9 AC	Open	96:00:2d:92:6b:00	-75.0	5180	100	9.78 s

3. ## JSON Response from /scanresults/

A. Another way of viewing the same information is to use the /scanresults/ URI. This URL can be found at your LANforge ip using port 8080. Ex: 192.168.10.20:8080/scanresults. We will also need the shelf number, the resource number, and the station name. The final URL would look like this **192.168.10.20:8080/scanresults/1/1/sta0000**

B. The scan results can be viewed through JSON by using cURL on the same URL as before. The response will look like this:

```
{"handler":"candela.lanforge.HttpStationScan$FixedJsonResponder","uri":"
scanresults/:shelf_id/:resource_id/:port_id","candela.lanforge.HttpStationScan":
{"duration":"1"},"scan-results":[{"1.1.4.08:36:c9:e3:d4:da":{"age":"2238","auth":"WPA2"
"beacon":"200","bss":"08:36:c9:e3:d4:da","channel":"44","entity id":"1.1.4",
"frequency":"5220","info":"3x3 MCS 0-9 AC","signal":"-32.0","ssid":"Logan-Test-Net"}}]}
```

4. ## Accessing and Printing JSON Response with Python

A. We will use sta_scan_test.py as an example for a `start()` method

A. First, we'll need to send a JSON post using realm. Usethis cookbook as reference for getting started with realm. Our JSON will look something like this:

```
data = {
"shelf": 1,
"resource": 1,
"port": self.sta_list
}
```

B. We can then use `json_post` to send the request. We'll need to wait about 15 seconds to give the scan time to happen

```
self.json_post("/cli-json/scan_wifi", data)
time.sleep(15)
```

C.

Next, we'll create a variable with the results from the scan using

```
scan_results = self.json_get("scanresults/1/1/%s" % ','.join(self.sta_list))
```

D. Finally, we'll create a loop to iterate through the JSON response and print some nicely formatted output

```
print("{0:<23}".format("BSS"), "{0:<7}".format("Signal"), "{0:<5}".format("SSID
for result in scan_results['scan-results']:
    for name, info in result.items():
        print("%s\t%s\t%s" % (info['bss'], info['signal'], info['ssid']))
```

B. Final Results

A. Our final function will look like this:

```
def start(self):
    self.station_profile.admin_up()
    print(self.sta_list)
    print("Sleeping 15s while waiting for scan")
    data = {
        "shelf": 1,
        "resource": 1,
        "port": self.sta_list
    }
    self.json_post("/cli-json/scan_wifi", data)
    time.sleep(15)
    scan_results = self.json_get("scanresults/1/1/%s" % ','.join(self.sta_list)

    print("{0:<23}".format("BSS"), "{0:<7}".format("Signal"), "{0:<5}".format("
    for result in scan_results['scan-results']:
        for name, info in result.items():
            print("%s\t%s\t%s" % (info['bss'], info['signal'], info['ssid']))
```

B. Our formatted output should look like this:

```
BSS                     Signal  SSID
00:0e:8e:52:4e:82       -33.0   test-net
08:36:c9:e3:d4:db       -31.0   Logan-Test-Net
08:36:c9:e3:d4:dc       -27.0   Logan-Test-Net
```