

## Record the results of a test as CSV from the REALM monitor script

**Goal:** Record the results of a LANforge test as a CSV file.

Some scripts in the LANforge library have a monitor function built in.

We are going to be using the `test_ipv4_variable_time` script for this demonstration.

This is useful for running a test and then analyzing the results afterwards.

- 
1. Start LANforge GUI. It is recommended to run this script on a fresh LANforge configuration with no stations loaded.
  2. Make sure you have lanforge-scripts on your device.  
If lanforge-scripts is already installed on your device, skip this step  
Navigate to py-scripts in the lanforge-scripts folder. If your LANforge device doesn't have this open source software yet you can clone them from [Github](#)  
To install lanforge-scripts paste `git clone https://github.com/greearb/lanforge-scripts` into your terminal.
  3. Type the following command into your command line  
`./test_ipv4_variable_time.py --radio wiphy0 --security wpa2 --ssid lanforge --password password --output_format csv`  
Replace the security, ssid, and password variables with the settings for the network you are testing.  
This will create 2 wiphy stations by default, connect them to the network you are testing, and report the results to a CSV file. You can change the following fields in the Realm Monitor function:
  4. This creates a default file in your report-data folder under your home directory. The name will be in the format with today's timestamp and the name of the test you ran. It's a normal Excel file which you can use however you want..
  5. There are multiple commands you can use with this function, here is a list of the flag and what each of them mean:
    - A. `report_file`: Name the full path of the file you want to save results to. Default will save to your report-data folder.
    - B. `duration_sec`: how long you want to run the test

- C. `output_format`: The output format you want your file in. The following formats are supported:
- A. `xlsx` DEFAULT
  - B. `pickle`  
HINT: `pickle` is recommended if you are going to be manipulating data in python since it preserves formatting and can be quickly loaded into a Pandas DataFrame without any manipulation required
  - C. `csv`
  - D. `json`
  - E. `pdf`  
WARNING: PDF is hard to export data from without an Adobe Acrobat license
  - F. `png`  
WARNING: `png` is going to export an image, do not use this if you are planning on manipulating your data because it does not preserve the numbers recorded
  - G. `html`
  - H. `hdf`
  - I. `parquet`
  - J. `stata`
- D. `ssid`: REQUIRED Name of the network you are connecting to
- E. `password`: REQUIRED Password to the network
- F. `radio`: REQUIRED The radio which you are going to create stations from.
- G. `security`: Match the security protocol of your router.
- H. `test_duration`: Default is 60 seconds, write in a any number if you need. You can also use minutes or hours notation in this command, so for 42 minutes write `42m` and for 8 hours write `8h`.
- I. `upstream_port`: Most users won't need to use this option, but it tells the program where to connect to the router
- J. `created_cx`: List of the cross connects you are going to be analyzing. If you are starting with no stations created, you won't need to use this option.