Notes for the Test Beam Setup at FermiLab - 2019 Edition

Martin

March 27, 2019

1 Computing Setup

1.1 Network Topology

The non-routable network 192.168.60.0/24 spans the control room and the enclosures. It is masqueraded (NAT'ed) via the 192.168.60.1 gateway machine, so machines on the network are able to get to the general internet. For a manual configuration, set up the network as follows:

IP address	192.168.60.X	
netmask	255.255.255.0	
broadcast	192.168.60.255	
gateway	192.168.60.1	
nameserver	131.225.0.254	FermiLab's Name server
ntp server	131.225.8.126/127	FermiLab's time server
ntp server	131.225.17.126/127	more time servers

These are the current devices (as of Mar 19, there will be more) connected to that network:

Name	IP address	Location	Role
eicdaq_fnal	192.168.60.111	enclosure	Main Data Acquisition Machine
web power switch	192.168.60.7	enclosure	power control
HV crate	192.168.60.55	enclosure	CAEN N1570 crate
CAEN HV Unit	192.168.60.56	enclosure	CAEN HV
cam1	192.168.60.80	enclosure	data logging camera pi
motorserver	192.168.60.81	enclosure	motor server
cam2 & 3	192.168.60.149	enclosure	data logging camera pi
"gecko terminal"	192.168.60.150	control room	HV control
ftbfbnl01local	192.168.60.1	rack in control room	FTBFBNL01 seen from our network

1.2 Accessing the Network from Outside

We made it so that we can log in from "outside" (that includes the general FermiLab network) to our DAQ machines. While Brookhaven uses ssh keys to authenticate users, FermiLab uses Kerberos to do the same. The downside for us is that everyone needs to use the kerberos software on his or her computer, but most of the time this is already installed.

Use the `kinit <username>` command to obtain a kerberos token. Remember that in most cases, especially on Windows, your username will not be the same as your login name at FermiLab. If successful, the `klist` command should show a *ticket granting ticket*:

```
$ klist
Ticket cache: KCM:501
Default principal: purschke@FNAL.GOV

Valid starting       Expires              Service principal
01/18/2017 13:36:14  01/19/2017 15:36:14  krbtgt/FNAL.GOV@FNAL.GOV
 renew until 01/25/2017 13:36:14
01/18/2017 13:36:36  01/19/2017 15:36:14  host/ftbfbnl01.fnal.gov@FNAL.GOV
 renew until 01/25/2017 13:36:14
```

Then you can login to our gateway machine (which is seen as 192.168.60.1 from the internal network) as user "ftbf_user" to ftbfbnl01.fnal.gov, as in

```
ssh -l ftbf_user ftbfbnl01.fnal.gov
```

Your FNAL account name must be enabled in order for you to login, which we can take care of ourselves. That ftbfbnl01 machine is one of ours, which got a new system disk where the computing division installed the standard Fermi SL6.

1.3 Making reasonable settings for logging in

The ftbfbnl01 gateway acts, in many aspects, like the rssh gateways at BNL. Consider setting up a script or an alias to log in consistently with the same tunnels (so you can bookmark some tunneled pages).

The services you may want to establish tunnels to are the Elog and maybe the cameras.

Try, as a suggestion (watch the last two lines, same ip, different ports)

```
ssh -l ftbf_user ftbfbnl01.fnal.gov \
    -L 17815:localhost:7815 \
    -L 10080:192.168.60.80:8081 \
    -L 10081:192.168.60.149:8081 \
    -L 10082:192.168.60.149:8082 \
    -L 10007:192.168.60.7:80
```

This allows you to see, in your local web browser

- the logbook as `http://localhost:17815`
- the first camera as `http://localhost:10080`.
- the second camera as `http://localhost:10081`.
- the third camera as `http://localhost:10082`.
- the power switch as `http://localhost:10007`.

You must add those lines to your $HOME/.ssh/config to enable the kerberos logins:

```
Host *.fnal.gov
  GSSAPIAuthentication yes
  GSSAPIDelegateCredentials yes
```

To make my life easier, I also added

```
Host eicdaq_fnal
     User eic
     ProxyCommand ssh  ftbf_user@ftbfbnl01.fnal.gov nc -w7200 %h %p

Host 192.168.60.*
     ProxyCommand ssh  ftbf_user@ftbfbnl01.fnal.gov nc -w7200 %h %p
```

so I can just type “ssh eicdaq_fnal” and am logged in as user “eic” on our DAQ machine.

The latter block allows me to log into any machine on our DAQ network directly; I can do

```
  ssh -l root 192.168.60.80
```

to go straight to the camera server.

1.4 Copying data from the daq machines to RCF

The obvious gateway into RCF is to rsync files to the proper destination on rftpexp.rhic.bnl.gov. Keep in mind that you need an ssh agent running (typically at the machine where the connection originates from, usually your laptop), which has *your* RCF key.

Here is how I copy (that’s from the T1044 days, replace appropriately):

```
rsync -av /data/eic/fnal/ purschke@rftpexp.rhic.bnl.gov:
      /sphenix/data/data03/phnxreco/sphenix/t1044/fnal/
```

(that’s all in one line.)

`rsync` is all about placing the right trailing slashes (or not), so the trailing slashes you see here are important.

Some of us experienced a lot of trouble with loggin in, which is not really different from our day-to-day interaction with the RCF.

Some pointers:

- you *must* run an ssh-agent on your laptop. Macs all do, Linux machines all do, on Windows machines you need to manually start one, either ssh-agent if you use cygwin, or “pagent” (comes as a putty add-on) if you are using putty.

- `ssh-add -L` is your friend. This command lists all "identities" that your agent has, and is the prime debugging tool if you think you should be able to login but cannot.

- verify that you can log in to RCF directly from your laptop. Once logged in, list the keys with `ssh-add -L`. Log in to the ftbfbnl01 machine and issue the `ssh-add -L` command there. It should show the same keys. If not, if it lists no identities, or displays a message that it cannot see an agent, your chain of ssh-agent forwards is broken. The agent connection gets forwarded through all the ssh hops from one machine to the next, which is set up on all our machines. If this is the case, the problem is virtually guaranteed to be with your laptop.

- have a look at `https://www.phenix.bnl.gov/WWW/offline/wikioff/index.php/Ssh_access_to_internal_machines_the_easy_way`, which explains most of the stuff that you should put into your `~/.ssh/config` file.