Contents

1 Creating a browser-based UI

One of the major advantages of using a single board computer is that it runs a full operating system, with existing libraries and software. In this lab, you will learn how to use that to create a browser-based interface through which users can interact with a Pi-based product.

We'll do this exercise in a virtual Linux environment.

1.1 By the end of this session...

You will be able to:

- Set up and access a virtual machine on the GENI testbed
- Create and run a basic Flask app that interacts with a *virtual* "HAT library"

(a HAT is a hardware circuit that is attached to a Pi - **H**ardware **A**ttached on **T**op).

In a future lab assignment, we'll run this browser-based interface on a Pi and use it to control a real circuit connected to the Pi.

1.2 Lab report

You will submit your lab work in Gradescope. You will upload some screenshots and answer some questions as described in the Gradescope assignment. You do not have to include anything else (e.g. no description of procedure, etc.)

1.3 Connect to a virtual Linux environment

We will complete this assignment using a Linux environment in a virtual machine, which you will access over SSH. This section will show you how to access a virtual Linux environment on the GENI testbed.

1.3.1 Prepare your GENI account

First, let's prepare your workstation. For SSH, you will need a terminal and an SSH client:

- If you are using a Mac or Linux device, the built-in terminal that comes with your operating system should include an SSH client. Make sure you know how to open the Terminal application that comes with your operating system. (Ask for help if you can't!)
- If you are using a Windows device and you do not already have an SSH client, you will have to download and install one. I recommend Git Bash, which you can download here: https://git-scm.com/downloads. Once you have downloaded and installed this application, make sure you know how to open a Git Bash terminal. (Ask for help if you can't!)

Once you have a terminal window open, run

```
ssh
```

You should see some usage information, like this (although not necessarily identical to this):

```
usage: ssh [-46AaCfGgKkMNnqsTtVvXxYy] [-B bind_interface]
           [-b bind_address] [-c cipher_spec] [-D [bind_address:]port]
           [-E log_file] [-e escape_char] [-F configfile] [-I pkcs11]
           [-i identity_file] [-J [user@]host[:port]] [-L address]
           [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]
           [-Q query_option] [-R address] [-S ctl_path] [-W host:port]
           [-w local_tun[:remote_tun]] destination [command]
```

This shows that you have an SSH client! If you aren't able to run an SSH client and see usage information, stop and ask for help.

Once you have your SSH client ready, you'll need to set up your account on GENI. GENI is a "virtual lab" for experiments on networking and distributed systems. It allows experimenters to set up *real* (not simulated!) end hosts and links at one or more GENI host sites located around the United States. Experimenters can then log in to the hosts asociated with their experiment to install software and run applications.

To set up your account, go to https://portal.geni.net and click on "Use GENI".

On the next screen, you will be prompted to choose an Identity Provider. Start typing the name of your identity provider, New York University, into the text input box. The field will start to suggest matching institutions after you type a few letters; when you see the name of your university appear underneath the text input box, click on it to select it, then choose "Continue". Then, when prompted, log in to your NYU account. (Your home institution username and password will not be sent to GENI; if you would like to read more about how this works, see InCommon Federation Basics.)

Review the policies on the next page, check both boxes, and click "Activate" to log in to the GENI Portal.

Next, you will set up a pair of SSH keys with which you will access resources on GENI. (If you already have a key pair set up on your laptop, you don't need to create new keys - you can skip to the part where you upload your public key to the GENI Portal.)

GENI users access resources using public key authentication. Using SSH public-key authentication to connect to a remote system is a more secure alternative to logging in with an account password.

SSH public-key authentication uses a pair of separate keys (i.e., a key pair): one "private" key, which you keep a secret, and the other "public". A key pair has a special property: any message that is encrypted with your private key can only be decrypted with your public key, and any message that is encrypted with your public key can only be decrypted with your private key.

This property can be exploited for authenticating login to a remote machine. First, you upload the public key to a special location on the remote machine. Then, when you want to log in to the machine:

1. You use a special argument with your SSH command to let your SSH application know that you are going to use a key, and the location of your private key. If the private key is protected by a passphrase, you may be prompted to enter the passphrase (this is not a password for the remote machine, though.)
2. The machine you are logging in to will ask your SSH client to "prove" that it owns the (secret) private key that matches an authorized public key. To do this, the machine will send a random message to you.
3. Your SSH client will encrypt the random message with the private key and send it back to the remote machine.
4. The remote machine will decrypt the message with your public key. If the decrypted message matches the message it sent you, it has "proof" that you are in possession of the private key for that key pair, and will grant you access (without using an account password on the remote machine.)

Of course, this relies on you keeping your private key a secret. Never share your private key with anyone else, and never post it online.

On your laptop, you're going to generate a key pair and upload the public key to the GENI portal. Then, you'll use that key from now on to log in to GENI resources.

If you are using Windows, download and install Git Bash and use its terminal. If you are using a Mac or Linux-based laptop, open a terminal.

Generate a key with:

```
ssh-keygen -t rsa
```

and follow the prompts to generate and save the key pair. The output should look something like this:

```
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/users/ffund01/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /users/ffund01/.ssh/id_rsa.
Your public key has been saved in /users/ffund01/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:z1W/psy05g1kyOTL37HzYimECvOtzYdtZcK+8jEGirA ffund01@example.com
The key's randomart image is:
+---[RSA 2048]----+
|                 |
|                 |
|           .  .  |
|          + .. . |
|    .   S .*.o  .|
|     oo. +ooB o .|
|    E .+.ooB+* = |
|        oo+.@+@.o|
|        ..o==@ =+|
+----[SHA256]-----+
```

In a safe place, make a note of:

- The passphrase you used,
- The full path to your private key (`/users/ffund01/.ssh/id_rsa` in the example above) - copy and paste this from your terminal output,
- The full path to your public key, which has the same name as your private key but with a `.pub` extension (`/users/ffund01/.ssh/id_rsa.pub` in the example above) - copy and paste this from your terminal output.

If you forget these, you won't be able to access resources on GENI - so hold on to this information!

Next, upload your public key to the SSH Keys section of your profile on the GENI portal: in the menu, click on your name, then on "SSH Keys", and upload your public key to that page. (Note that your public key is the one with the `.pub` file extension.)

This step may be tricky - some students may have trouble finding their key to upload it. If you can't upload your key, ask for help!

GENI users are organized into projects (read more here.) Anyone can create an account on GENI, but unless you are part of a project (supervised by a responsible Project Lead), you won't be able to access any GENI resources or run experiments.

In the GENI Portal, click on "Home > Projects" in the menu at the top, then click "Join a Project". Type the project name

into the box where it says “Enter a Project Name”, click “Join”, and proceed to send the join request.

Requests to join a project are pending until they are approved by the Project Lead (your instructor!), so it may take a few minutes for your request to be approved. You’ll receive email notification when that happens.

To do anything on GENI, you need to create a *slice*. A slice is a “container” for the resources that will be used in an experiment (read more here).

Slices have *members*; anyone who has is a member of a slice will have their keys installed on all resources that have been reserved since they were added as a member. As project lead, your instrutor has access to all resources on all slices in the project. When you create a slice, you will have access to the resources in your slice, as long as you don’t lose your key!

Create a slice by clicking New Slice in the portal: then enter a slice name and click “Create Slice”. All slices in a project must have a unique name. To make your slices easy to distinguish from your classmates’, please include your username in your slice name. For example, for this assignment, I might call my slice “lab1-ff524.”

Slices expire. That’s OK - they are supposed to expire if you’re finished with an experiment. But pay special attention to your slice expiration date! When your slice expires, you will lose access to all of the resources in your slice. Individual resources have their own expiration date, which may be different than, but not later than, the slice expiration date. You must make sure to retrieve any data saved on your resources before they expire.