Connecting to target

shan@iha.dk
V1.6

Introduction

In this exercise you will get the target (DevKit8000) set up so that you can communicate with it from the host. You will connect the target to the host, log in on the target and transfer a file via the USB connection.

Prerequisites

In order to complete this exercise, you must:

- Have completed the *Getting set up* exercise, thus a working Linux (in a VMWare)
- Have access to a USB-to-serial-converter or have a serial port on your PC
- Ensured that the DevKit8000's SD card image is up to date. If not, follow the instruction found in *Guide to manipulating the DevKit8000 image*[1].

Exercise 1 Connect to the target

Connect your target to your PC and thus your Kubuntu in its VMware image.

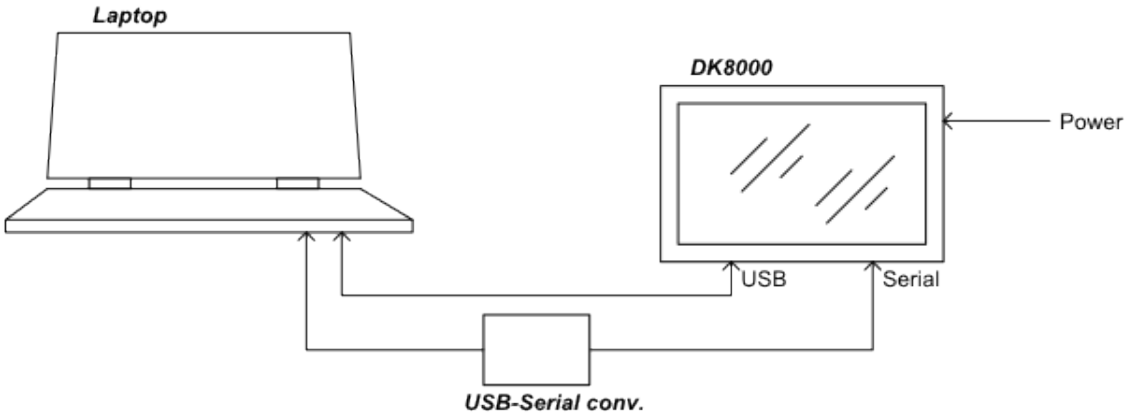

Figure 1.1: How Host is connected to Target

In figure 1.1 host and target can be connected by two different means. One is via a serial connection and the other is via a network connection that works via the USB subsystem. The later is also called tethering.

In this exercise we will focus on tethering. Upon having connected the two as described above, run the following command in a terminal: `sudo ifup usb0`.

[1] Can be found here: `https://redmine.iha.dk/devs/projects/devkit8000/wiki/Wiki`

Connecting to target

Exercise 2 Test connection

Connect to the target using *Secure Shell* (SSH) - see the Wiki[2] for details on how to use SSH. Remember that you log in as `root` on the DevKit8000.

Do note that your VMware has the IP address 10.9.8.1 and the target has the IP address 10.9.8.2. These addresses are hard coded into both host and target, but only active when the USB cable is connected between the two.

If you get the error message `ssh: connect to host 10.9.8.2 port 22: No route to host`, your USB interface is probably down. Check the following:

- The USB cable is connected between host and USB
- The Netchip RNDIS/Ethernet Gadget Device is connected in the Kubuntu host (Go to "VM → Removable Devices" to check this)
  Do note that there known problems with USB3. If you therefore experience problems make sure that you not using a USB3 port
- Verify that the kernel has detected a USB device having something resembling a MAC address. You can checkout the kernel log quick & simple by running `dmesg`[3]
- The USB interface is up. In an Kubuntu shell, type `ifconfig usb0`. If the device is not *UP*, type `sudo ifup usb0`.
- Sometimes the usb0 interface may be up without an IP. If uncertain take down the interface by typing `sudo ifdown usb0` followed by type `sudo ifup usb0`.
- If you are warned about a "MAN-IN-THE-MIDDLE-ATTACK" then run the command: `ssh-keygen -R 10.9.8.2`[4].

Now try to establish the connection again.

Exercise 3 Move a file

Create a small file, e.g. with the command `touch abc` (creates the empty file `abc`). Then use *Secure Copy* `scp` to move it to the target. Check that the file is actually present on the target. Use the *man-pages* if you need help on how it works.

Exercise 4 Create shell scripts for ssh and scp (optional)

Albeit optional but very convenient to have completed.

The commands for `ssh` and `scp` are repetitive and tedious. Create two small shell scripts as follows:

[2] `http://www.osoffice.de/howto/an-easy-how-to-use-ssh-and-scp-for-linux-beginners.html` otherwise JFGI. Likewise if you do not wish to specify an empty password each time you connect to target see `http://linuxproblem.org/art_9.html`

[3] You should be acquainted with this command from exercise 1

[4] Every ssh server generates an unique ID upon first start-up. The client saves a given server's IP and ID, if the server changes ID then the ssh client interprets this as being a "MAN-IN-THE-MIDDLE-ATTACK". Changing embedded devices that all have the same IP, effectively changes the ID associated with an IP, thus making the client believe that something bad has happened.

Connecting to target

- `conn2tgt` should establish an SSH connection to the DevKit8000 on IP address 10.9.8.2 (the USB interface)
- `cp2tgt [file]` should copy the file/folder `file` to the DevKit8000

Checkout the program `sshpass`, it is very useful when/if you have a target that has a password, as you can pass the password to `ssh` or `scp` via command line options to `sshpass`.

If its not installed use the command `sudo apt-get install sshpass`.