Advanced Network Security Lab- Session 0: Requirements

Dominik Roy George
d.r.george@tue.nl

Savio Sciancalepore
s.sciancalepore@tue.nl

January 30, 2024

1 Introduction

In the course 2IMS30, namely, Advanced Network Security, we provide three Lab sessions. The sessions involve the usage of a real Operative System (OS) for IoT devices (*contiki-ng*), as well as real IoT devices (*Sensortag CC2650*). To fully enjoy lab sessions, it is essential for the students to have configured their machines beforehand correctly. This document provides the required instructions for configuring the Lab Environment. We kindly request the students to follow the instructions. Otherwise, having the Lab Environment not ready will cause delays.

☞ We remark that we only provide support for Virtualbox and not for any other hypervisor products.

2 Pre-Requisites for Lab Sessions

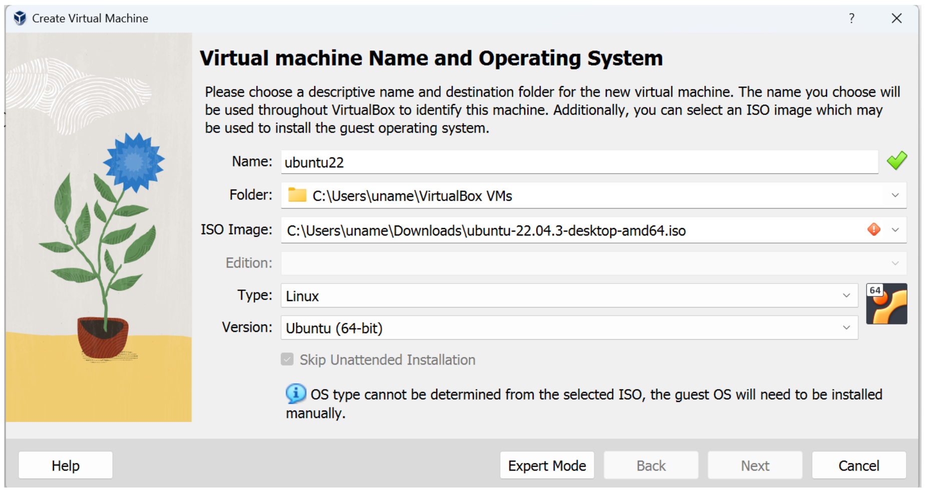
To fully enjoy lab sessions of the course, you need to pre-configure the following main blocks: (i) Ubuntu Virtual Machine; (ii) Necessary Packages; (iii) ARM Compiler, (iv) UNIFLASH, and (v) Contiki-ng. You will be working with a Virtual machine, necessary to compile the firmware of contiki-ng, while the flashing will be happening through your host Windows machine, where UNIFLASH is installed.

2.1 Step 1: Ubuntu Virtual Machine

First install Virtualbox. After installing the Virtualbox, download the image of Ubuntu-22.04.3-desktop-amd64.iso.

During the installation check the check box for: **"skip unattended installation"**(1):

After having Ubuntu OS fully installed in Virtualbox, install the Guest Additions of Virtualbox. Therefore, follow the instructions from this blog::


Create Virtual Machine

Virtual machine Name and Operating System

Please choose a descriptive name and destination folder for the new virtual machine. The name you choose will be used throughout VirtualBox to identify this machine. Additionally, you can select an ISO image which may be used to install the guest operating system.

Name: ubuntu22

Folder: C:\Users\uname\VirtualBox VMs

ISO Image: C:\Users\uname\Downloads\ubuntu-22.04.3-desktop-amd64.iso

Edition:

Type: Linux

Version: Ubuntu (64-bit)

Skip Unattended Installation

OS type cannot be determined from the selected ISO, the guest OS will need to be installed manually.

Figure 1: Skip unattended installation.

`linuxtechi.com/install-virtualbox-guest-additions-on-ubuntu/` to install the Virtualbox Guest additions.

Thereafter, reboot the Virtual Machine (VM) and create a shared folder. The shared folder is necessary to copy the compiled executable so you can access that folder from the host machine to flash them on the devices. Create a folder on your Windows host machine such as: *ans/assignments* and follow the steps here to create a shared folder in the VM: `https://carleton.ca/scs/tech-support/troubleshooting-guides/creating-a-shared-folder-in-virtualbox/`. While creating a shared folder, check the properties: *auto mount and permanent*

'After adding the shared folder, execute the following command in the VM to access it with your current user (do not execute command in sudo su session):

Listing 1: Compile

```
$ sudo adduser $USER vboxsf
```

Afterwards, restart the VM and follow the instructions below and type in your terminal.

2.2 Step 2: Necessary Packages

After having fully installed and configured such OS in Virtual Box and the Guest additions, you can install the following packages by using a terminal and typing the following commands:

Listing 2: Package installation for Ubuntu

```
$ sudo apt update && sudo apt upgrade -y && sudo apt install -y
    cmake make build-essential doxygen git git-lfs curl net-tools
     wireshark srecord rlwrap autoconf automake libxmu-dev gcc-
    msp430 default-jdk ant openjdk-11-jdk python3-pip python3-
    serial libncurses-dev p7zip-full libusb-dev
```

☞ During the installation of wireshark a prompt will appear if you choose yes, then type following command:

```
$ sudo usermod -a -G wireshark $USER
```

2.3 Step 3: ARM Compiler

After installing the packages in the VM, the next step is to install the ARM Compiler on the VM. Apply the following instructions to install the ARM compiler:

Listing 3: ARM Compiler Installation

```
$ mkdir ~/toolchain && cd ~/toolchain && wget https://armkeil.
    blob.core.windows.net/developer/Files/downloads/gnu-rm/9-2020
    q2/gcc-arm-none-eabi-9-2020-q2-update-x86_64-linux.tar.bz2 &&
     sudo tar -xjf gcc-arm-none-eabi-9-2020-q2-update-x86_64-
    linux.tar.bz2 -C /usr/share/

$ sudo ln -fs /usr/share/gcc-arm-none-eabi-9-2020-q2-update/bin/
    arm-none-eabi-gcc /usr/bin/arm-none-eabi-gcc

$ sudo ln -fs /usr/share/gcc-arm-none-eabi-9-2020-q2-update/bin/
    arm-none-eabi-g++ /usr/bin/arm-none-eabi-g++

$ sudo ln -fs /usr/share/gcc-arm-none-eabi-9-2020-q2-update/bin/
    arm-none-eabi-gdb /usr/bin/arm-none-eabi-gdb

$ sudo ln -fs /usr/share/gcc-arm-none-eabi-9-2020-q2-update/bin/
    arm-none-eabi-size /usr/bin/arm-none-eabi-size

$ sudo ln -fs /usr/share/gcc-arm-none-eabi-9-2020-q2-update/bin/
    arm-none-eabi-objcopy /usr/bin/arm-none-eabi-objcopy

$ sudo ln -fs /usr/lib/x86_64-linux-gnu/libncurses.so.6 /usr/lib/
    x86_64-linux-gnu/libncurses.so.5

$ sudo ln -fs /usr/lib/x86_64-linux-gnu/libtinfo.so.6 /usr/lib/
    x86_64-linux-gnu/libtinfo.so.5
```

2.4 Step 4: UNIFLASH

The following step is to download and install UNIFLASH, which is a program to flash the Sensortag CC2650 with the executables compiled for ARM. The installation of UNIFLASH is straightforward by following the instructions of the UNIFLASH Windows installer. Download UNIFLASH on your Windows host machine.

2.5 Step 5: Contiki-NG

The last step is to clone the Contiki-ng repository and initiate the submodules in the VM.

Listing 4: Contiki-NG

```
$ git clone https://github.com/contiki-ng/contiki-ng.git && cd
    contiki-ng && git submodule update --init --recursive
```

After applying all the steps successfully, you should have an environment where it is possible to compile executable for the Sensortag CC2650. To check if the steps were successful, you can navigate in contiki-ng folder to the examples and compile a program:

Listing 5: Compile

```
$ cd /your-working-folder/contiki-ng/examples/hello-world
$ sudo make TARGET=cc26x0-cc13x0 BOARD=sensortag/cc2650
```

After executing the command you should have hello-world.cc26x0-cc13x0 executable.

The instructions are based on the Contiki-NG Documentation [1].

References

[1] C. NG, "Contiki linux toolchain." [Online]. Available: https://docs.contiki-ng.org/en/develop/doc/getting-started/Toolchain-installation-on-Linux.html