Practical Work

Tool installation

Benoît FEIX & Jérémy DRON

Abstract

For the labs (TPs), we will need various tools. To make their installation easier, we have prepared a Docker environment for you. However, it is still possible to manually install each tool one by one.

Prerequisites :
- Install git
- Install docker
- Install docker-compose

Keywords: python, jupyter, numpy, matplotlib, side-channel

1. Installation

1. Create a folder named *TP_SCA*

   -
     ```bash
     mkdir TP_SCA; cd TP_SCA
     ```
2. Place the *dockerfile* in the TP_SCA folder
3. Create the Docker image using the dockerfile.

   -
     ```bash
     docker build -t cryptis-cw .
     ```

     (Don't forget the "." at the end of the command)
   - The installation may take several minutes
4. Start your Docker image to verify everything works
   - Follow the steps in the **"Start the Docker Image"** section, then return here
5. A folder named *my-notebooks* has been created. Set yourself as the owner:

   -
     ```bash
     who am i
     ```

     (retrieves your *user_name*)

   -
     ```bash
     chown <user_name> my-notebooks
     ```

   -
     ```bash
     ls -la
     ```

     (verifies ownership change)
6. Go to the *my-notebooks* folder and clone the repository with the labs (TPs)

-
  ```bash
  cd my-notebooks
  ```

-
  ```bash
  git clone <url-des-TPs> .
  ```

  (Don't forget the "." at the end of the command)

2. Start the Docker Image

To start your Docker, there are several options. Choose the easiest one for you!

2.0.1. Docker-compose (Recommended)

1. Get the *docker-compose.yml* file and place it in the same location as the *dockerfile*.
2. Start your Docker by following the instructions in the *docker-compose.yml* file

   -
     ```bash
     docker compose up
     ```
3. Docker will launch in the terminal, opening Jupyter Lab
   - Click on the link in your terminal starting with https://127.0.0.1/

2.0.2. bash Command

1. Start the Docker image

   -
     ```bash
     sudo docker run --privileged --device=/dev/bus/usb:/dev/bus/usb -it -p 8888:8888 -v ./my-notebooks:/opt/my-notebooks cryptis-cw
     ```
2. Docker will launch in the terminal, opening Jupyter Lab
   - Click on the link in your terminal starting with https://127.0.0.1/

You can also add the command to your *.bashrc* to create an alias:

1. Add the command to your *.bashrc*

   -
     ```bash
     alias cwrun='sudo docker run --privileged --device=/dev/bus/usb:/dev/bus/usb -it -p 8888:8888 -v ./my-notebooks:/opt/my-notebooks cryptis-cw'
     ```
2. Reload your shell

   -
     ```bash
     source ~/.bashrc
     ```
3. Start Jupyter Lab

   -
     ```bash
     cwrun
     ```

2.0.3. .sh File

1. Create a *run_cw.sh* file containing your command commande

   -
     ```sh
     #!/bin/bash
     sudo docker run --privileged --device=/dev/bus/usb:/dev/bus/usb -it \
         -p 8888:8888 \
         -v ./my-notebooks:/opt/my-notebooks \
         cryptis-cw
     ```

2. Allow script execution
   -
     ```bash
     chmod +x run_cw.sh
     ```
3. Start Jupyter Lab
   -
     ```
     bash ./run_cw.sh
     ```
4. Docker will launch in the terminal, opening Jupyter Lab
   - Click on the link in your terminal starting with https://127.0.0.1/

3. Project Structure

The project is made up of several folders described below:

- **my-notebooks** – Interface folder between your machine and Docker. It contains the TP notebooks. In your filesystem, it is located in the TP_SCA folder created during installation.[1]
- **SideSCA-Traces-Public** – Contains all the traces needed for the TPs, stored in .ets format, easily usable with estrace.
- **chipwhisperer** – GitHub repository containing everything needed to use the ChipWhisperer-Lite. Includes:
  - **firmware (mcu)** – Contains the C code of all firmware we will use. They will be cross-compiled and uploaded to the CW-Lite.
  - **chipwhisperer-jupyter** – Example notebooks provided by ChipWhisperer, which you can test
  - **Setup_scripts** – Notebooks to easily connect the hardware to our attack programs
  - **software** – Python library code for ChipWhisperer

3.1. Open a Terminal Inside Docker

A terminal is available in Jupyter, but it is not very user-friendly. You can open one directly from your own terminal.

1. Retrieve the Docker container ID
   -
     ```bash
     docker ps
     ```

     (the ID is under CONTAINER ID)
2. Execute a new shell inside the Docker container
   -
     ```bash
     docker exec -it <ID> /bin/bash
     ```

3.2. Delete Docker Image

⚠ **WARNING: Deleting the Docker image may cause you to lose your work!**

1.
   ```bash
   docker rmi cryptis-cw
   ```

[1] Only this folder is in both your filesystem AND Jupyter; the others exist only inside Jupyter.