ChipWhisperer: How to use the Jupyter environment

Samuele Yves CERINI

Research Fellow - CINI Cybersecurity National Laboratory

samuele.cerini@consorzio-cini.it

CYBER CHALLENGE.IT

CYBERSECURITY NATIONAL LABORATORY

https://cybersecnatlab.it

License & Disclaimer

License Information

Disclaimer

- We disclaim any warranties or representations as to the accuracy or completeness of this material.
- Materials are provided “as is” without warranty of any kind, either express or implied, including without limitation, warranties of merchantability, fitness for a particular purpose, and non-infringement.
- Under no circumstances shall we be liable for any loss, damage, liability or expense incurred or suffered which is claimed to have resulted from use of this material.

Prerequisites

- Lectures:
  - *HS_3.1 – Side Channel Attacks*

Goal

- Presenting the Jupyter environment used to interact with the ChipWhisperer boards
- Installing the environment on your PC
- Using the environment to complete the interactive tutorials on side-channel analysis

Outline

- Introduction and prerequisites
- Installing Docker
- Building the container (i.e., the Jupyter environment)
- Accessing the tutorials
- URL, Port and User token
- Uploading the challenges to the Jupyter environment
- Stopping + removing the environment from your PC

Introduction and prerequisites

- The use of a GNU/Linux distribution is assumed
  - Windows or MacOS are technically supported, but their use has not been tested nor is it suggested
  - In case you don't have Linux, a Virtual Machine (with internet access) is sufficient

Introduction and prerequisites

- The Jupyter environment we are going to use is packaged as a Docker container, to ease deployability and avoid the so-called "dependency hell"
- **To use it, we need to:**
  - Download a zip archive, containing all the files necessary to build the environment
  - Install Docker, if not already present on your PC
  - Build the container, using the dockerfile provided
  - Run the container and launch the Jupyter server (on localhost)
  - Access the Jupyter tutorials via browser

Downloading the `client.zip` file

- Reach your CyberChallenge personal page
  - Click on the "Challenges" tab
  - Reach the "Hardware Security 3" module
  - Open the first challenge "HS_3.00 - Jupyter Tutorials Environment"
  - Download the `client.zip` file from the attachments
  - Extract the content of the archive on your PC
- The folder you extracted contains:
  - A "client-tutorials-dockerfile" file, used by Docker to build the environment

Downloading the zip: screenshots

Challenges - CyberChal

https://ctf.cyberchallenge.it/challenges

499 — 26 — 3
SS_3.01 - ReallyOptimizedPrimality test
software
pwn ROP remote

500 — 8 — 0
SS_3.04 - Try-your-luck
software
pwn BOF PIE ASLR remote

Hardware Security 3

0 — 6 — 0
HS_3.00 - Jupyter Tutorials Environment
hardware

Access Control

384 — 422

Challenges - CyberChal | Training materials - Cyb

https://ctf.cyberchallenge.it/challenges#challenge-143

499
SS_3.01 - ReallyOptimizedPrimality test
software
pwn ROP remote

500
SS_3.04 - Try-your-luck
software
pwn BOF PIE ASLR remote

Hardware Security 3

0
HS_3.00 - Jupyter Tutorials Environment
hardware

Access Control

384
AC_1.01 - Bootstrap
access control

HS_3.00 - Jupyter Tutorials Environment

6 — 0 — 0

hardware

Download the "client.zip" attachment and follow the given instructions to install the Jupyter environment. The instructions are provided in the slides accessible in the "Training" tab, second lecture.
Below, we give you the URL of the CyberChallenge.it server, the port of the service and your personal user token (keep it secret!)
URL: 123.456.789.012
PORT: 12345
TOKEN: "H3r3_15_Y0uR_T0k3n"

Attachments

client.zip

Installing Docker

- **On Fedora (recommended):**
  - How to install Docker on Fedora
- **On Ubuntu (recommended):**
  - How to install Docker on Ubuntu
- On Windows (NOT recommended):
  - How to install Docker on Windows
- On MacOS (NOT recommended):
  - How to install Docker on MacOS

Building the Jupyter environment

- To build the container with the Jupyter environment:
  1. Open a terminal **in the `client` folder**;
  2. Run the following to build the Docker image:
     - `docker build --pull -f client-tutorials-dockerfile -t cc_tutorials:latest .`

Building Jupyter: screenshot

Activities Terminal apr 4 13:24

Home Downloads client

Recent
Starred
Home
Documents
Downloads
Music
Pictures
Videos
Trash
Ubuntu 21.1...
Other Locations

cyberchallenge_client
tutorials
client_requirements.txt
client-tutorials-dockerfile

sem@sem-boxes: ~/Downloads/client

```
sem@sem-boxes:~/Downloads/client$ docker build --pull -f client-tutorials-dockerfile -t cc_tutorials:latest .
Sending build context to Docker daemon  9.695MB
Step 1/19 : FROM ubuntu:20.04
20.04: Pulling from library/ubuntu
4d32b49e2995: Pull complete

Digest: sha256:bea6d19168bbfd6af8d77c2cc3c572114eb5d113e6f422573c93cb605a0e2ffb
Status: Downloaded newer image for ubuntu:20.04
 ---> ff0fea8310f3
Step 2/19 : ENV TZ=Europe/Rome
 ---> Running in 3fe35cb9c69a
Removing intermediate container 3fe35cb9c69a
 ---> 288c73fe93e7
```


Starting the Jupyter environment

- To start the container you built:
  1. Issue the following:
     - `docker run --name 'cc_tutorials' -ti -p 8888:8888 cc_tutorials:latest`

```
Removing intermediate container 227087b63511
 ---> b8b1a2908ff3
Successfully built b8b1a2908ff3
Successfully tagged cc_tutorials:latest
sem@sem-boxes:~/Downloads/client$ docker run --name 'cc_tutorials' -ti --security-opt no-new-privileges -p 8888:8888 cc_tutorials:latest
[I 13:26:43.778 NotebookApp] Writing notebook server cookie secret to /home/cc_tutorials/.local/share/jupyter/runtime/notebook_cookie_secret

[I 13:26:44.217 NotebookApp] Serving notebooks from local directory: /home/cc_tutorials
[I 13:26:44.218 NotebookApp] Jupyter Notebook 6.4.7 is running at:
[I 13:26:44.218 NotebookApp] http://2c68ae35ac34:8888/?token=...
[I 13:26:44.218 NotebookApp]  or http://127.0.0.1:8888/?token=...
[I 13:26:44.218 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
```

Accessing the tutorials

- To access the tutorials:
  1. Follow the instructions printed on terminal, or...
  2. Open a web browser and type:
     - `127.0.0.1:8888`
  3. Enter the token "cc_tutorials" to access the notebooks
     1. Only the first time, from now on Jupyter will use a cookie to remember your login
  4. Open the "tutorials" folder and access the notebook "00-Intro.ipynb"
  5. Follow the tutorials in order
  6. Enjoy!

Accessing the tutorials: screenshot

Activities | Firefox Web Browser | apr 4 13:30

Home | Downloads | client

Recent
Starred
Home
Documents
Downloads
Music
Pictures
Videos
Trash
Ubuntu 21.1...
Other Locations

cyberchallenge_client | tutorials | client_requirements.txt | client-tutorials-dockerfile

sem@sem-boxes: ~/Downloads/client

```
Step 14/19 : RUN mkdir /home/$USER_NAME/tutorials
 ---> Running in 1d7bee80eed9
Removing intermediate container 1d7bee80eed9
 ---> 58e65a2e7956
Step 15/19 : COPY --chown=$USER_NAME:$USER_NAME ./tutorials/tutorials /home/$USER_NAME/tutorials/
 ---> a6faa3580840
Step 16/19 : COPY --chown=$USER_NAME:$USER_NAME ./tutorials/LICENSE
 ---> d29c01fe1688
Step 17/19 : RUN rm /home/${USER_NAME}/client_requirements.txt
 ---> Running in c58579826883
Removing intermediate container c58579826883
 ---> 40039aec57c1
Step 18/19 : RUN rm -rf /home/${USER_NAME}/cyberchallenge_client
 ---> Running in 09e4f09320d3
Removing intermediate container 09e4f09320d3
 ---> 022d48656f39
Step 19/19 : CMD jupyter notebook --port=8888 --no-browser --ip=0.0
 ---> Running in 227087b63511
Removing intermediate container 227087b63511
 ---> b8b1a2908ff3
Successfully built b8b1a2908ff3
Successfully tagged cc_tutorials:latest
sem@sem-boxes:~/Downloads/client$ docker run --name 'cc_tutorials'
[I 13:26:43.778 NotebookApp] Writing notebook server cookie secret

[I 13:26:44.217 NotebookApp] Serving notebooks from local directory:
[I 13:26:44.218 NotebookApp] Jupyter Notebook 6.4.7 is running at:
[I 13:26:44.218 NotebookApp] http://2c68ae35ac34:8888/?token=...
[I 13:26:44.218 NotebookApp]  or http://127.0.0.1:8888/?token=...
[I 13:26:44.218 NotebookApp] Use Control-C to stop this server and
[I 13:28:22.089 NotebookApp] 302 GET /?token= (172.17.0.1) 0.990000
[I 13:28:22.282 NotebookApp] 302 GET /tree?token= (172.17.0.1) 1.20
```

Jupyter Notebook

127.0.0.1:8888/login?next=%2Ftree%3Ftoken%3D

jupyter

Password or token: •••••••••••• Log in

cc_tutorials

Token authentication is enabled

If no password has been configured, you need to open the notebook server with its login token in the URL, or paste it above. This requirement will be lifted if you enable a password.

The command:

```
jupyter notebook list
```

will show you the URLs of running servers with their tokens, which you can copy and paste into your browser. For example:

```
Currently running servers:
http://localhost:8888/?token=c8de56fa...
:: /Users/you/notebooks
```

or you can paste just the token value into the password field on this page.

See the documentation on how to enable a password in place of token authentication, if you would like to avoid dealing with

URL, Port and User Token

- As you will find explained in the introductory notebooks, the interactive tutorials "talk" to a remote ChipWhisperer board, connected to our servers
- To access it, you need the following:
  - The URL/IP address of the CyberChallenge.it server
  - The port of the service talking to the ChipWhisperer board
  - Your personal user token, used to identify your request
    - Keep it secret! Don't disclose it to your colleagues!

URL, Port and User Token

- Where to find them?
  - You can find URL, port and personal token in the same page from which you downloaded the `client.zip` archive: follow the same instructions
- How to use them?
  - In some tutorials, you will be asked to fill the URL, port and token parameters in the respective code snippets, see next slide

URL, Port and User Token

Activities Firefox Apr 6 13:08

Home Downloads client

Recent
Starred
Home
Documents
Downloads
Music
Pictures
Videos
Trash
Fedora-WS-L...
Other Locations

tutorials/ | 01-Learning to fly - Jupyte

127.0.0.1:8888/notebooks/tutorials/01-Learning to fly.ipynb 110%

jupyter 01-Learning to fly (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

Run Markdown

As for now, I just need you to *roughly* understand what is happening here.

In [ ]:
```
# Import the CyberChallenge library
from cyberchallenge_client import ccclient
```

The `URL`, `PORT` and `YOUR_TOKEN` information are provided to you in your cyberchallenge.it personal portal. Don't change them, of course.

In [ ]:
```
# Create a "connection" object
# Use the Shift+Tab shortcut to open the completion hints (and the documentation)
# YOU MUST RUN THE PREVIOUS SNIPPET to have a working autocompletion pop-up
connection = ccclient.Utility(str(URL), int(PORT), str(YOUR_TOKEN))
```

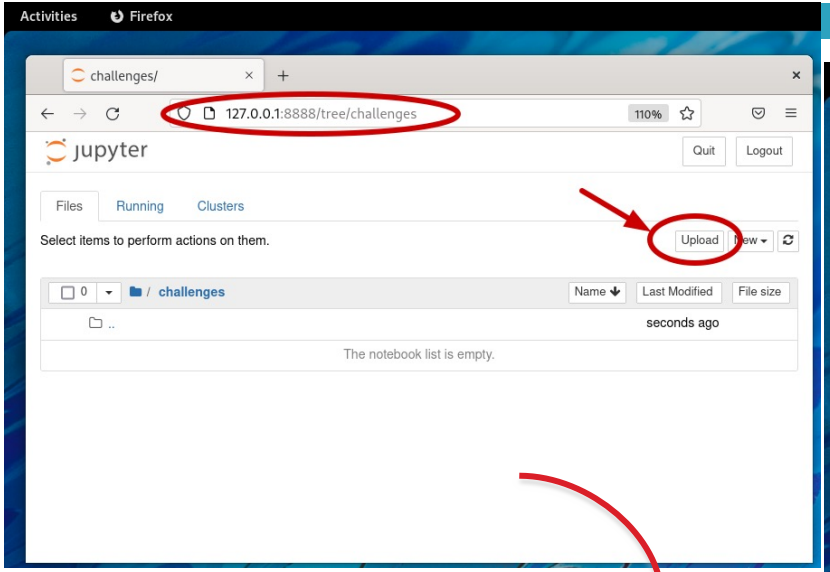
In [ ]:
```
# Launch a capture request, with this command we are actually connecting to the server and the
# Use the Shift+Tab shortcut to open the completion hints (and the documentation)
# YOU MUST RUN THE PREVIOUS SNIPPET to have a working autocompletion pop-up
# Let's capture a single trace
ccclient.default_capture_config["num_traces"] = 1
(state, project) = connection.capture_request("firestarter", 60, ccclient.default_capture_conf
```

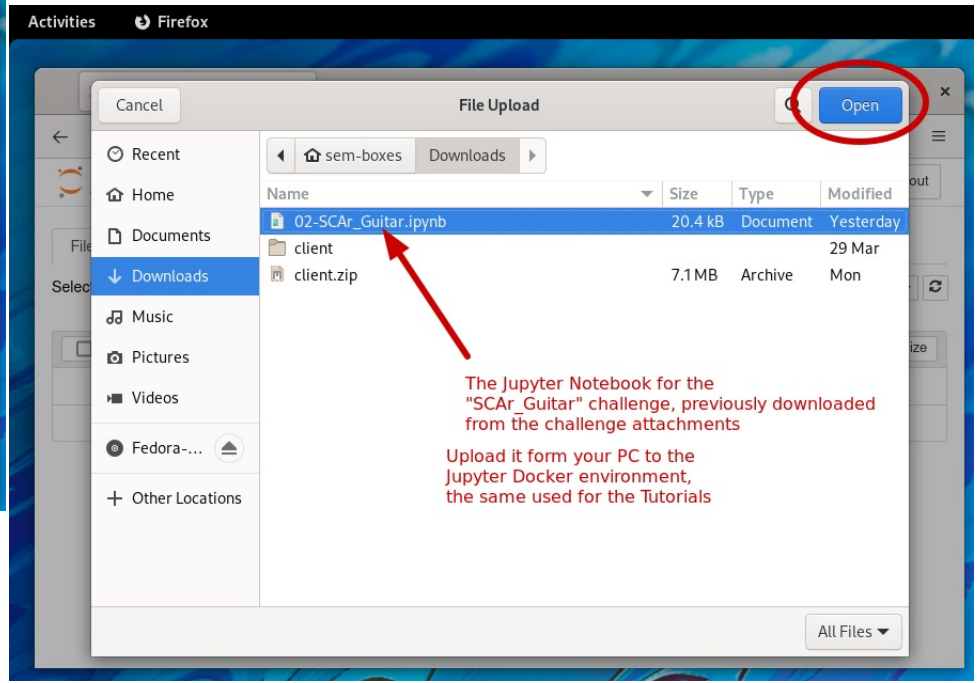
And that's it, if everything worked well, the `state` variable should be `True` and your captured traces should be saved in the `project` object.

Uploading the challenges

- How to complete the related challenges?
  - Reach your CyberChallenge personal page, click on the "challenges" tab
  - Select a challenge from "HW Security 3", read it and download the "attachment" file
    - The file is a Jupyter Notebook
- I have the file, what happens now?
  - Launch the same docker environment you used for the tutorials
  - Upload the file inside the folder "challenges"
  - Open it and complete the challenge, good luck!
    - To access the ChipWhisperer board, use the exact same functions we used in the tutorials
    - URL, Port and User Token are provided in the challenge page

Uploading the challenges: screenshots





Stopping the Jupyter environment

- 2 ways:
  1. Ctrl-C on the opened terminal
     - Only for the first time you launched Jupyter
  2. With the following command
     - From the second time onwards
     - `docker stop cc_tutorials`
- How to restart a stopped container:
  1. Assuming you already launched it once (`docker run`)
     - `docker start cc_tutorials`

(Before) Removing the environment

⚠️ **DATA LOSS WARNING** ⚠️

- Removing the container hosting the Jupyter environment WILL DELETE ALL YOUR PROGRESS!
  - Removing the container deletes all the tutorials/challenges and any additions you may have made to them (text, graphs, code snippets etc...)
- You may want to save the tutorials/notebooks on your PC before removing the environment, see next slide

Saving your progress

```
 ---> d29c01fe1688
Step 17/19 : RUN rm /home/${USER_NAME}/client_requirements.txt
 ---> Running in c58579826883
Removing intermediate container c58579826883
 ---> 40039aec57c1
Step 18/19 : RUN rm -rf /home/${USER_NAME}/cyberchallenge_client
 ---> Running in 09e4f09320d3
Removing intermediate container 09e4f09320d3
 ---> 022d48656f39
Step 19/19 : CMD jupyter notebook --port=8888 --no-browser --ip=0.0.0.
 ---> Running in 227087b63511
Removing intermediate container 227087b63511
 ---> b8b1a2908ff3
Successfully built b8b1a2908ff3
Successfully tagged cc_tutorials:latest
sem@sem-boxes:~/Downloads/client$ docker run --name 'cc_tutorials' -t
[I 13:26:43.778 NotebookApp] Writing notebook server cookie secret to

[I 13:26:44.217 NotebookApp] Serving notebooks from local directory: /
[I 13:26:44.218 NotebookApp] Jupyter Notebook 6.4.7 is running at:
[I 13:26:44.218 NotebookApp] http://2c68ae35ac34:8888/?token=...
[I 13:26:44.218 NotebookApp]  or http://127.0.0.1:8888/?token=...
[I 13:26:44.218 NotebookApp] Use Control-C to stop this server and shu
[I 13:28:22.089 NotebookApp] 302 GET /?token= (172.17.0.1) 0.990000ms
[I 13:28:22.282 NotebookApp] 302 GET /tree?token= (172.17.0.1) 1.2000
[I 13:30:28.934 NotebookApp] 302 POST /login?next=%2Ftree%3Ftoken%3D
[I 13:30:45.725 NotebookApp] Writing notebook-signing key to /home/cc_
[W 13:30:45.728 NotebookApp] Notebook tutorials/00-Intro.ipynb is not
[I 13:30:45.951 NotebookApp] 302 GET /notebooks/tutorials/img/CCIT_Log
[I 13:30:46.293 NotebookApp] 302 GET /notebooks/tutorials/img/00-Intro
[I 13:30:47.014 NotebookApp] Kernel started: 20dd39f8-bb19-4187-b327-
```

Ctrl-C in the terminal will shutdown (stop) the container, but it will not remove it.
You don't risk any data loss in this case.

To restart the container, simply run "docker start cc_tutorials"

Remember to repeat this procedure for every notebook you wish to save on your PC!

Removing the environment

⚠️ **DATA LOSS WARNING** ⚠️

- Removing the container hosting the Jupyter environment WILL DELETE ALL YOUR PROGRESS! (see previous slides)

- To remove the docker container:
  - `docker rm cc_tutorials`
- ⚠️ The command DOES NOT ask you for confirmation ⚠️

ChipWhisperer: How to use the Jupyter environment

Samuele Yves CERINI

Research Fellow - CINI Cybersecurity National Laboratory

samuele.cerini@consorzio-cini.it

CYBER CHALLENGE.IT

CYBERSECURITY NATIONAL LABORATORY

https://cybersecnatlab.it