

UP23 Lab01

Date: 2023-03-06

- UP23 Lab01
- Build course environment
 - Objective
 - Instructions
 - Grading

Build course environment

Objective

This lab aims to build a runtime environment required by this course. You have to be familiar with `docker` and `pwntools` in this lab. Please follow the instructions to complete this lab. Once you have completed any grading item, please demo it to the TAs.

Instructions

1. Prepare your own docker environment. You can install Docker Desktop (<https://www.docker.com/products/docker-desktop/>) on your laptop or simply use the `docker.io` package pre-installed in the classroom desktop PC.
2. Download the `docker-compose.yml` (<https://people.cs.nctu.edu.tw/~chuang/courses/unixprog/resources/ubuntu/docker-compose.yml>) and `Dockerfile` (<https://people.cs.nctu.edu.tw/~chuang/courses/unixprog/resources/ubuntu/Dockerfile>) from the course website.

For Apple Chip Users (M1/M2): You have to enable "Use Docker Compose V2" in your Docker Desktop options and use the alternative Dockerfile file here (<https://people.cs.nctu.edu.tw/~chuang/courses/unixprog/resources/ubuntu/m1/Dockerfile>).
3. Build your docker container environment. Ensure that you have correctly set up your username and created the home directory for the user.

You must use your own user/group name in the docker instead of the built-in default name.

4. Follow the instructions in the introduction slide, compile textbook samples, and run in your container instance.
5. Install `pwntools` by following the instructions here (<https://md.zoolab.org/s/ElETCdAQ5>).
6. Once `pwntools` is installed successfully, please solve the challenge running at

```
nc up23.zoolab.org 10363
```

Note that there is a `pow` challenge before you can actually solve the challenge. Please read the `pow` (proof-of-work) (<https://md.zoolab.org/s/EHSmQ0szV>) document first.

7. The challenge asks you to solve big number mathematics and use base64 to encode the result in a shortest little endian represented binary number (8-bit aligned because one byte has 8 bits). For example, given the equation $108713406511 * 137993468292$, the numeric result is `15001740014290984849212`. Based on the challenge requirement, you should send `PE+hYV/09j4tAw==`, which is encoded from a byte sequence containing `['0x3c', '0x4f', '0xa1', '0x61', '0x5f', '0xf4', '0xf6', '0x3e', '0x2d', '0x03']`.

Grading

1. [10pts] Prepare your own runtime environment (Linux OS running on dockers, VMs, or physical machines). Please ensure that you are using your own name instead of `chuang`.
2. [10pts] Ensure that your (external) files are accessible in your runtime docker (or VM). If you run Linux natively on a physical machine, you can skip this step and get the 10pts automatically.
3. [10pts] Install `pwntools` and ensure that the following script works in the Python3 interpreter.

```
from pwn import *
r = process('read Z; echo $Z', shell=True)
r.sendline(b'AAA')
r.interactive()
```

4. [65pts] Solve the challenge by implementing a `pwntools` script.

