UP24 Lab01

Date: 2023-02-26

Build course environment

Objective

This lab aims to build a runtime environment required by this course. You have to be familiar with `docker` , `python` , 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.

> You ***MUST*** uplaod your scripts to E3 before you demo.

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/debian/docker-compose.yml) and Dockerfile (https://people.cs.nctu.edu.tw/~chuang/courses/unixprog/resources/debian/Dockerfile) from the course website.

   > ***For Apple Chip Users (M1/M2)***: You have to enable “Use Docker Compose V2” in your Docker Desktop options.

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 ***simple HTTP challenge*** by implementing a `pwntools` script to retrieve IP address from the URL: http://ipinfo.io/ip (http://ipinfo.io/ip). You may try to play with the URL using the command:

   ```
   curl http://ipinfo.io/ip
   ```

   Your script output should be equivalent to the above command. To see the details about how `curl` interacts with the remote server, you may pass an optional parameter `-v` to `curl`.

7. Please also solve the challenge running at

   ```
   nc up.zoolab.org 10681
   ```

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

   The challenge asks you to solve simple mathematical equations. You have to read the equations and send the answer back to the server. Each of the equations is encoded in the base64 encoding. You have to decode the equations by your script, calculate the answer, and then send the answer to the server.

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. [20pts] Solve the ***simple HTTP challenge*** described above.
5. [50pts] Solve the challenge running on our challenge server.

FAQ

- Please check the port number you connected if you modify the `pow.py` directly.
- Make sure your home directory is writable by the user you created. (You could use `chown` or `chmod` to change the directory ownership or permission.)
- There is no `\n` at the end of each challenge, so you will get an exception when reading the challenge’s math expression with `recvline()` .
- The interacting environment tells you the number of challenges at the beginning.