

Compiling and running the RISC-V Test

Step 1(Tool Chain setup in Ubtunu VM):

Download RISC-V GNU Tool Chain with

```
$ git clone https://github.com/riscv/riscv-gnu-toolchain
```

cd into the file via the terminal

configure the RISC-V GNU Tool Chain run:

```
$ sudo apt-get install autoconf automake autotools-dev curl python3  
python3-pip libmpc-dev libmpfr-dev libgmp-dev gawk build-essential  
bison flex texinfo gperf libtool patchutils bc zlib1g-dev  
libexpat-dev ninja-build git cmake libglib2.0-dev
```

Install and link the newlib:

```
./configure --prefix=/opt/riscv
```

Run the Makefile with:

```
make
```

Step 2(Make arbitrary c file):

Here's a sample:

main.c

```
#include <stdio.h>  
#include <string.h>  
int main(int argc, char *argv[]){  
    printf("Congratulations on getting your first flag!!\n");  
}
```

Step 3(Compile the c file):

Compile the c file with:

```
riscv64-unknown-elf-gcc -o main.elf main.c
```

****Note the output file must be a .elf file to be run directly as a challenge****

Step 4(Create a challenge GitHub repo):

1. Create a dojo.yml file:

Sample:

```
id: example
//Challenge name
name: Example Dojo

description: |
  This is an [example dojo](https://github.com/pwncollege/example-dojo).
  Fork this repository, and create your own dojo!

type: example
//module name and challenge name
modules:

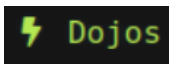
- id: main
  name: main
  description: The first module in this example dojo.

challenges:
- id: main
  name: main
  description: This is main.
```

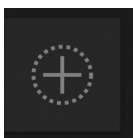
2. Create a directory folder
3. Store the compiled c file in the directory folder
4. Push to GitHub

Step 5(Adding & Running the challenges):

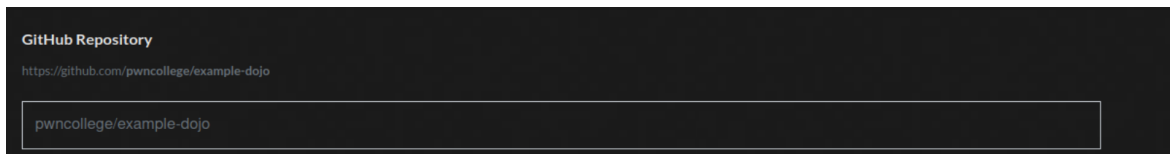
1. Go to <http://pwniot.cacti.academy/>
2. Login as admin (username: admin password: admin)
3. Click Dojos



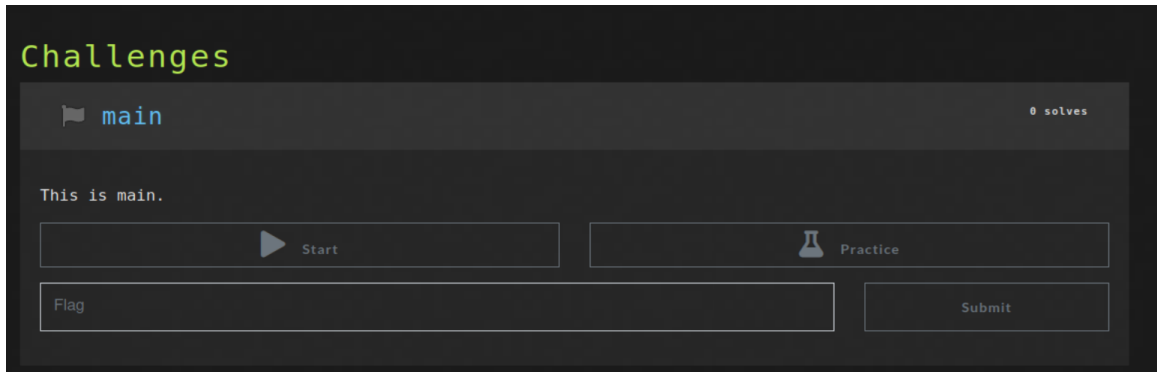
4. Click the (+) button



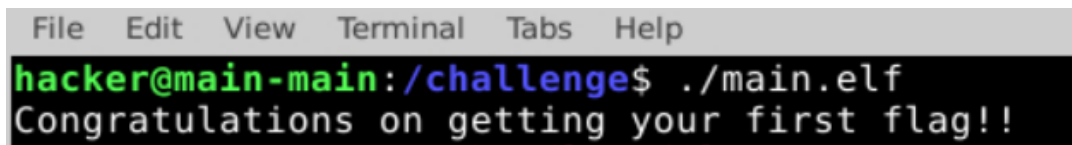
5. Add your github repo



6. Click the corresponding challenges and modules
7. Start the challenge



8. Go to Workspace(Vscode) or Desktop(VM). Pick the one you prefer
 9. Run `cd /challenge`
 10. Do ls to see a list of challenges
 11. Run the challenge using `./challenge_name.elf`
 12. Check the corresponding output
- Ex:



Sample RISC-V Challenge Repo:

<https://github.com/AnUnknownStranger/AnUnknownStranger-RISCV>